



G's ACADEMY
TOKYO

PaintApp



アジェンダ

❖ inputの基礎 (color, range...)

❖ Canvasの基礎

本日の演習

❖ 課題

本日の授業内容

- input属性 (color,range)
- Canvas基礎
- ペイントアプリ作成

input
color,range

input要素

color



date

年 / 月 / 日

range



text

text

input要素 1

<input type="text" name="項目名" value="値">

input要素 type属性	型	
type="text"	テキストボックス	改行なしテキスト
type="email"	テキストボックス	メールアドレス
type="password"	テキストボックス	パスワード
type="number"	テキストボックス	数値入力
type="radio"	ラジオボタン	複数の選択肢から1つを選択
type="checkbox"	チェックボックス	複数の選択肢から複数選択可能
type="submit"	サブミットボタン	form内の入力データを送信
type="hidden"	テキストボックス(非表示)	

input要素 2

<input type="text" id="項目名" value="値">

input要素 type属性	型	
type="date"	カレンダー	日付選択or日付入力
type="color"	カラーボックス	カラー選択orカラー入力
type="range"	スライダー	min,maxで最小値と最大値を設定
type="number"	テキストボックス (ステップ付)	数値のみ入力:min,maxで設定可能
type="file"	File選択ボタン	ファイルを選択することが可能

今日はこの color , range を
使ってみましょう！

input_color.html

span#viewを最初から用意してるので、
HTML/SCRIPTを追加して#viewに表示してみましょう！

```
12 <h1>選択した色 : <span id="view"></span></h1>
13 色選択 <input type="color" id="color">
14
15 <script>
16   $("#color").on("change",function(){
17     const color = $("#color").val();
18     $("#view").html(color);
19   });
20 </script>
```

input_range.html

```
12 <h1>線の太さ : <span id="view"></span></h1>
13 線の太さ選択 : <input type="range" id="range" min="1" max="10" value="1">
14
15 <script>
16   $("#range").on("change",function(){
17     const range = $("#range").val();
18     $("#view").html(range);
19   });
20 </script>
```

Canvas

canvasの基礎

【 canvas要素を知ろう 】

<canvas>要素の属性は主に 3 つ !

- ID
- width
- height

記述例 :

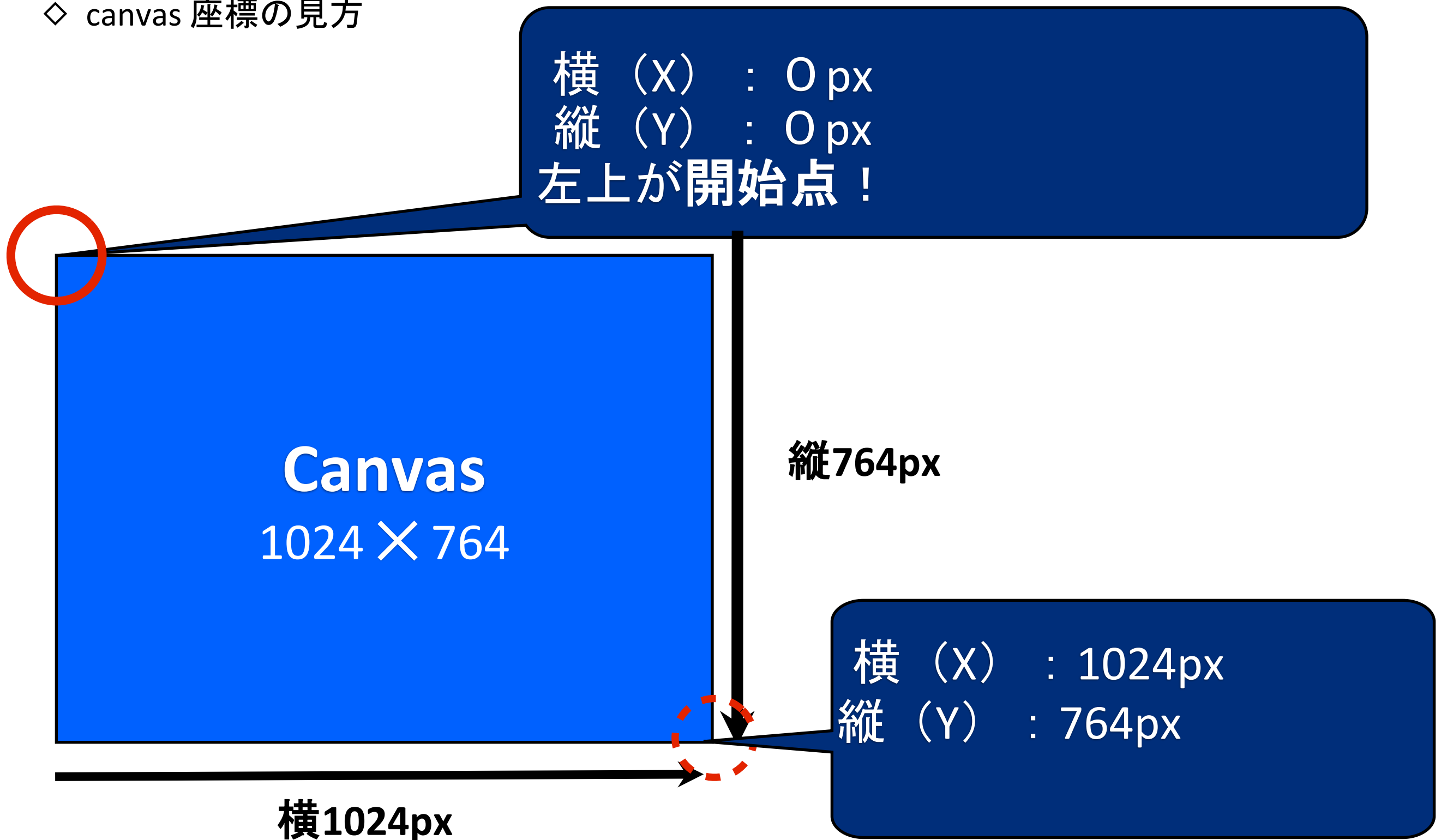
```
<canvas id=" * *" width="1024" height="764"></canvas>
```

```
=====
id=" * *" → ユニークID名を記述
width = "横幅を記述"
height = "縦幅（高さ）を記述"
=====
```

◇注意 : サイズ属性のサイズ指定はstyle属性では使用できません !

canvasの基礎

◇ canvas 座標の見方



canvasの基礎

【 canvasで描画準備 】

canvas 要素は描画機能にアクセスするための "getContext" と呼ばれるDOM メソッドを持っています。

"getContext" は描画コンテキストという1つの引数しか取ることができません。

現在利用可能なコンテキストは "2d" コンテキストの1つです。

※今後は2D以外に3Dも利用できる日が来るかもしれません。

記述例 :

```
const can = $("#drowarea")[0]; //jQueryのセレクタを使用
```

```
const ctx = can.getContext("2d");
```

最初の行では getElementById メソッドを使って DOM ノードを取り出しています

。

そして "getContext("2d")" メソッドを使って描画コンテキストにアクセスできます。

【 canvasで文字を描画 】

サンプル
canvas_font.html

取得したコンテキストで文字を描画をおこないます

```
//カラー指定
ctx.fillStyle = "#ff0000";
//fontサイズ、書式
ctx.font = "20px _sans";
//文字の設置位置
ctx.textBaseline = "top";
//表示文字と座標
ctx.fillText("Hello World!", 100, 80 );
```

ctx.textBaseline: 文字の設置するベースライン

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

canvasの基礎

【 canvasで矩形を描画 】

取得したコンテキストで矩形を描画します

```
ctx.rect(0,0,60,60);  
ctx.stroke();
```

サンプル
canvas_sikaku.html

サンプル
canvas_sankaku.html

指示形態は以下の通りです
Rect(座標、幅と高さ)

◆塗りつぶす場合

```
ctx.fillStyle = "#000";  
ctx.rect(10,10,40,40);  
ctx.fill();
```

塗りつぶす場合は**fill**を使用します。

必ず色をつける場合は「**fill**」と指定してください

canvasの基礎

サンプル
canvas_arc.html

【 canvasで円を描画 】

◇取得したコンテキストで円を描画します

```
ctx.arc(100, 100, 50, 0, Math.PI*2, false);
```

◇指示形態は以下の通りです

arc(座標、半径、円のスタート度、エンド度(描画)、回転)

◇塗りつぶす場合

```
ctx.fillStyle = "#000";  
ctx.arc(100, 100, 50, 0, Math.PI*2, false);  
ctx.fill();
```

塗りつぶす場合はfillを使用します。

必ず色をつける場合は「fill」と指定してください

【 canvasで線を描画 】

//context. beginPath()が呼ばれると初期化され始点は座標(0,0)へ

```
ctx.beginPath();
```

// x と y は新しい始点の座標となります。

```
ctx.moveTo(①開始横軸, ②開始縦軸);
```

// x と y は"線の終点の座標"となります。

```
ctx.lineTo(③終了横軸, ④終了縦軸);
```

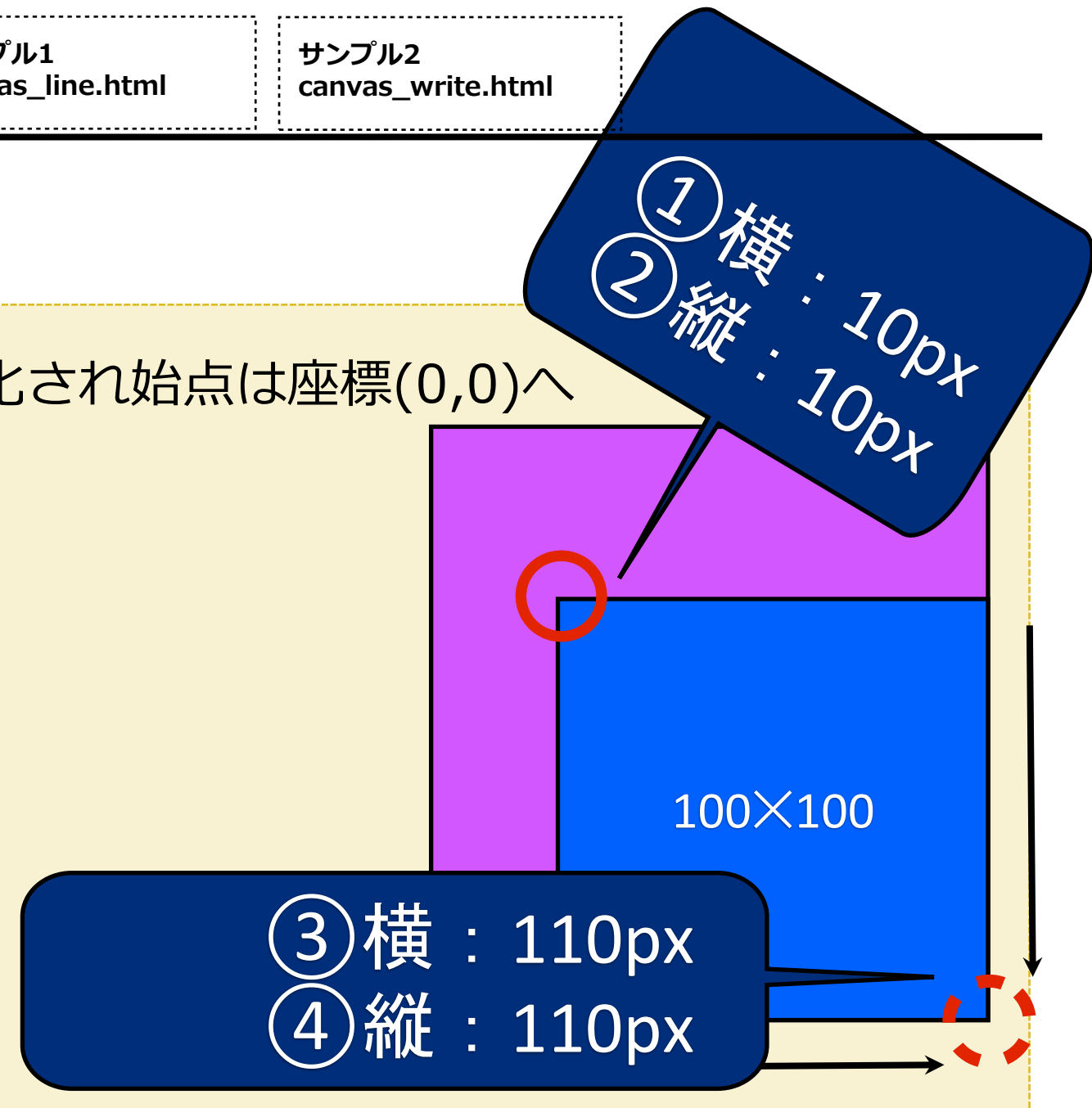
//線を描く

```
ctx.stroke();
```

//現在の点から始点に向けて直線を描くことで図形を閉じようとしています。

※もし図形がすでに閉じられていれば関数は何もしません。

```
ctx.closePath();
```



授業演習

ペイントアプリ作成 (考え方を学ぶ授業)

[sample/canvas_base.html](#)

●演習：canvas base.html

◇ お絵かきアプリ作成では
3つのイベントを作成！！

1. mousedown → 描くことをSTARTさせる
2. mousemove → 描き続ける
3. mouseup → 描くことをSTOPさせる

このアプリではどのような操作でアプリが動作するのかを明確にしましょう！
ここが重要POINT！！

上記が最低ラインとして制作

javascriptの基礎

【 関数 : 変数スコープ 】

スコープ範囲にも 2 つの種類があります

- ・グローバル変数 : 全ての領域から参照が可能
- ・ローカル変数 : { ... }内だけで参照が可能

```
<script>
//グローバル
const global = "グローバル";      //グローバル宣言
function scopeCheck() {
    const local = "ローカル"; //ローカルで宣言
    console.log("確認 1 : "+ global); //表示
}

scopeCheck(); //関数実行
console.log("確認 2 : "+ local);    //エラーになる

</script>
```

※関数内でconst,let を使って変数宣言しない場合、グローバル変数を参照します。

課題発表

●課題 kadai_canvas.html (canvas_base.htmlをコピーして作成)

◇ お絵かきアプリ作成

1. 「canvas要素内でmousedown+mousemove中は線を描く」 ※授業中で作成
2. 「mouseupでマウス放したら線を描かないようにする」 ※授業中で作成
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
5. 線の太さを変える

チュータリングタイム中に完了してください！！

上記が最低ラインとして制作

● プラスαでアイデアがわからない場合

canvas base.html

◇ アイデアなければ以下など

1. 保存ボタン → LocalStorage保存 (toDataURL)

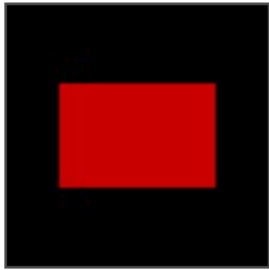
2. LocalStorage保存したデータを表示

※ toDataURL (canvas を画像に変換)

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

例)

<http://venezia-works.com/canvas/canvas3.html>



data: url の中身

```
data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAASABIAAD/4QBMRXhpZgAATU0AKgAAAAGAAAYdpAAQAAAABAAAAAGgAAZ
```

コード

```
/* canvasに赤色の矩形を描画する */
ctx.fillStyle = "rgb(200, 0, 0)";
ctx.fillRect(20, 30, 60, 40);
/* canvasの描画結果をPNGで取り出しimg要素にセット */
try {
  var img_png_src = canvas.toDataURL();
  document.getElementById("image_png").src = img_png_src;
  document.getElementById("data_url_png").firstChild.nodeValue = img_png_src;
} catch(e) {
  document.getElementById("image_png").alt = "未対応";
}
/* canvasの描画結果をJPEGで取り出しimg要素にセット */
try {
  var img_jpeg_src = canvas.toDataURL("image/jpeg");
  document.getElementById("image_jpeg").src = img_jpeg_src;
  document.getElementById("data_url_jpeg").firstChild.nodeValue = img_jpeg_src;
} catch(e) {
  document.getElementById("image_jpeg").alt = "未対応";
}
```

canvasのデータを文字で取得

👉 [toDataURL\(\)](#) メソッドの説明に戻る

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

発想が大事

こんなこともできます

- <http://venezia-works.com/animals/>
※CanvasとCSS3
- <http://venezia-works.com/file/index.html>
- <http://venezia-works.com/file/index2.html>
※Canvasと画像ライブラリ。
- <http://venezia-works.com/canvas/canvas.html>
※リアルタイム通信とCanvas組み合わせ

課題作成のPOINT

- 最小機能のアプリを先に完成させる
(ロジックがシンプルで、早い段階で動くアプリを確認できる)
- 小さいアプリを完成 → それから 1 機能を追加
(シンプルなロジックに 1 機能追加するので考えやすい)
- デザイン & コードにこだわるのは最後の作業！
- この機能何に使う？ → この機能を使えばこんな
の作れるな！！と発想を膨らませる練習
(世の中に無いアプリを作っていく世界なので重要なスキル)
- ラーニングシステムでの予習と復習