G's ACADEMY
TOKYO

ジーズアカデミー講座

本日の授業内容

Canvas基礎

ペイントアプリ作成

Canvas(2d)

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 ”

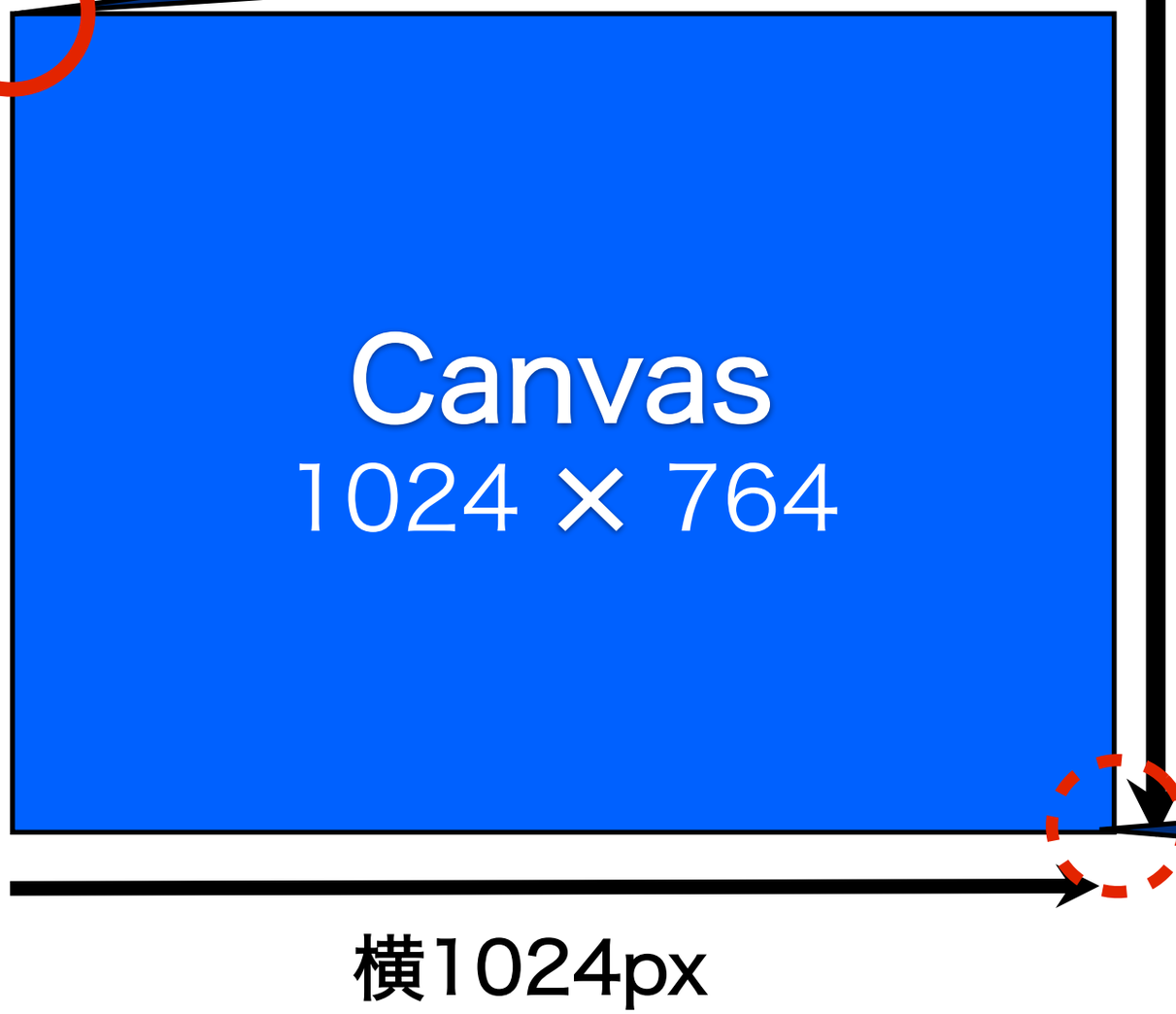
==========

◇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

横（X）：0px
縦（Y）：0px
左上が開始点！



横（X）：1024px
縦（Y）：764px

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。
ーー

記述例：

```
var can = document.getElementById("drowarea");
var 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。
ーー

canvasの基礎

【 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 】

サンプル
canvas_sikaku.html

サンプル
canvas_sankaku.html

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();
```

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

【 canvasで円を描画 】

サンプル
canvas_arc.html

◇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

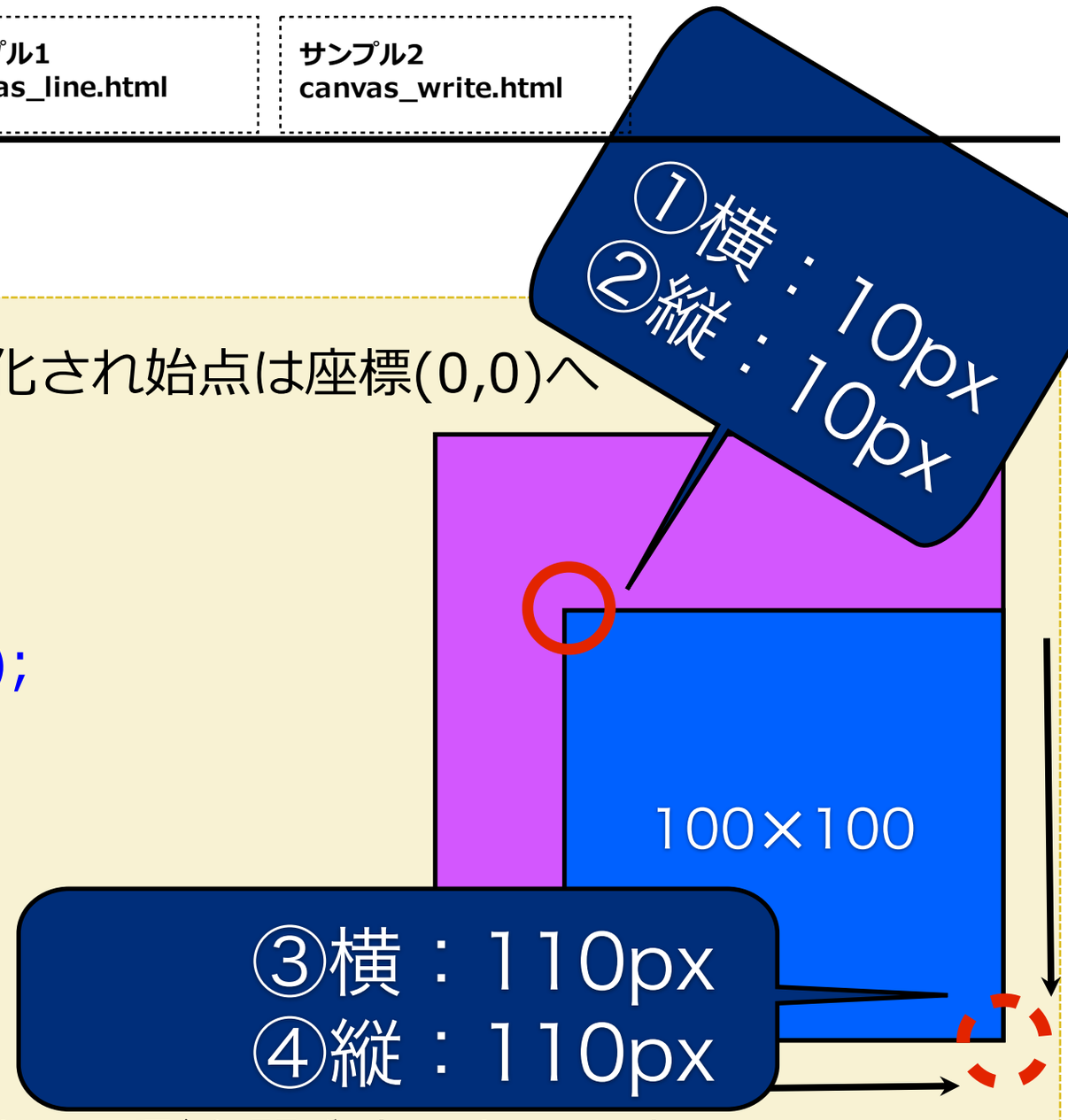
canvasの基礎

サンプル1
canvas_line.html

サンプル2
canvas_write.html

【 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
context.beginPath();
// x と y は新しい始点の座標となります。
context.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。
context.lineTo(③終了横軸, ④終了縦軸);
//線を描く
context.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。
context.closePath();
```



課題演習

１．◯が右枠着いたら→◯が左に戻る
右左を行ったり来たりさせる！
sample/ move kadai.html

２．グラフィックアートに挑戦
好きな様に作る！！
好きなファイル作成してどうぞ！

授業演習

ペイントアプリ作成
(考え方を学ぶ授業)

sample/canvas.html

●授業中課題

---

canvas.html

◇ お絵かきアプリの処理概要

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

上記が最低ラインとして制作

課題発表

●次回課題

---

kadai_canvas.html

◇ お絵かきアプリの処理概要
１．線の太さを変える
２．保存ボタン → LocalStorage保存 (toDataURL)
３．LocalStorage保存したデータを表示

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

上記が最低ラインとして制作

JSON
- データ形式 -

javascriptのJSON

※サンプル：json.html

【 JSON（JavaScript Object Notation）】

JavaScriptのオブジェクト表記法を元にした「データ形式」。変数や配列などのデータを、PHP/JAVA/他言語とデータ通信する際に”文字列”にして通信します。
（最近ではどの言語でもJSONを扱う関数が用意されています。）

```
var obj=[
        { fname:"Tarou" , lname:"Yamazaki" },
        { fname:"Anna" ,  lname: "Smith" },
        { fname:"Peter" ,  lname: "Jones" }
];
var json_text = JSON.stringify(obj);    //オブジェクトからJSONに変換
 console.log(json_text);                //コンソールにて表示確認
```

変数 obj を、
JSON形式の文字列に変換

変数 json_text には、以下文字列（JSON形式）に変換され代入されます。

```
[
   {"fname":"Tarou","lname":"Yamazaki"},
   {"fname":"Anna","lname":"Smith"},
   {"fname":"Peter","lname":"Jones"}
]
```

JSON形式の文字列

※サンプル：json.html

ではナゼ JSON は必要なのか？～通信の流れ～

パソコン:ブラウザ

json_text

①変数

```
[
  {"fname":"Tarou","lname":"Yamazaki"},
  {"fname":"Anna","lname":"Smith"},
  {"fname":"Peter","lname":"Jones"}
]
```

②JSON形式の文字列

③JSON文字列を送信

サーバ：PHP/JAVA

④JSON文字列を受信

⑤変数

json_text

```
[
  {"fname":"Tarou","lname":"Yamazaki"},
  {"fname":"Anna","lname":"Smith"},
  {"fname":"Peter","lname":"Jones"}
]
```

変数では他の言語にデータを渡せない、JSON形式の”文字列”に変換することで可能になる！

javascriptのJSON

※サンプル：json.html

【 JSONデータ（文字列からオブジェクトに変換）

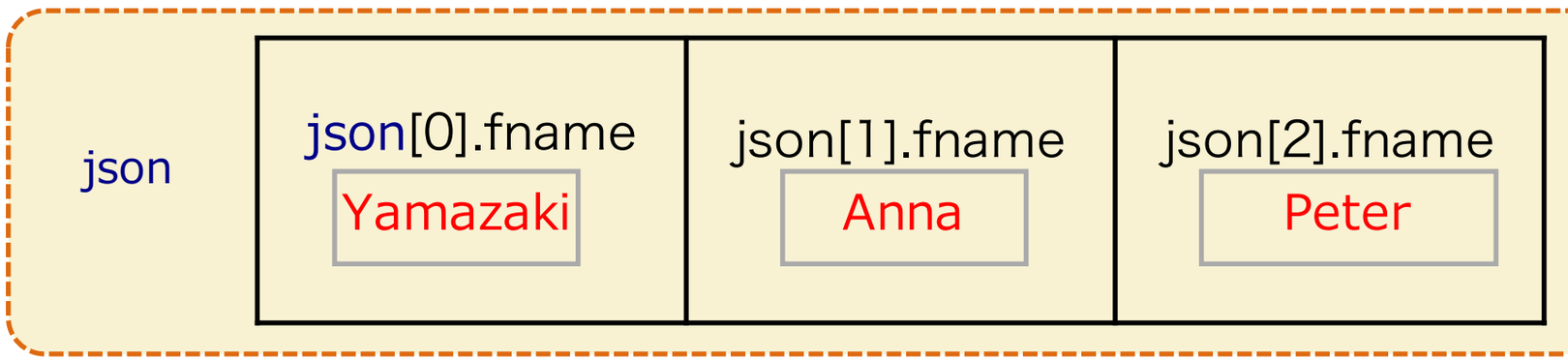

```
[
  {"fname":"Tarou","lname":"Yamazaki"},
  {"fname":"Anna","lname":"Smith"},
  {"fname":"Peter","lname":"Jones"}
]
```

JSON形式の文字列

```
var json = JSON.parse(json_text); //JSONからオブジェクトに変換
console.log(json);
```

json_text の中は以下の文字列（JSON形式）

【 オブジェクトへのデータの参照 方法例】



練習1 オブジェクトとJSONの相互変換

サンプル
json1.html

```
//問題文[連想配列/オブジェクト]
var quest = {
    qs: {
        q1: "通るときには閉まって、通らないときには開いているものは何？",
        q2: "話すことがとても好きな道具は何？",
        q3: "世界の真ん中にいる虫は何？"
     }
};
//JSONからオブジェクトに変換
 var json_text = JSON.stringify( quest);
$("body").append( json_text );

 //オブジェクトからJSONに変換
 var object = JSON.parse( json_text );
//jQueryの”append”を使って文字列を”Bodyタグ内に追加”
$("body").append("<p>"+ object.qs.q1 +"</p>");
$("body").append("<p>"+ object.qs.q2 +"</p>");
$("body").append("<p>"+ object.qs.q3 +"</p>");
```

変換後の表示を確認！