

memopad

Contents

- 配列
- オブジェクト
- localStorage
- localStorageの練習(メモ帳初級編)
- JSON & メモ帳中級編
- 課題発表 -> P2Pタイム

rules...

- 授業中は常にエディタを起動！
- 考えたことや感じたことはslackのガヤチャンネルでガンガン発信！
- 質問はslackへ！ 他の人の質問にも目を通そう！（同じ質問があるかも）
- 演習時，できた人はスクショなどslackに貼ってアウトプット！
- まずは打ち間違いを疑おう！
 - `{'";` など
- 書いたら保存しよう！（よく忘れる！）
 - `command + s`
 - `ctrl + s`

本日のゴール

- いろいろなデータ形式を知る！
- データの扱いになれる！
- データを保存・読み出しする！

配列

配列とは？？（ハマリポイントその2）

```
// 配列 (array)  
//   - 複数の値に順番をつけてまとめて扱う方法. 奥（闇）が深い.  
//   - 順番を「インデックス」と呼ぶ. 「0」からスタート！
```

```
// 例
```

```
<script>  
  const array = ['大吉', '中吉', '小吉', '凶', '大凶'];  
  alert(array[0]); // 大吉  
</script>
```



配列とは？？(ハマリポイントその2)

// 配列の定義

```
const list01 = ['月', '火', '水', '木', '金', '土', '日'];  
const list02 = new Array('月', '火', '水', '木', '金', '土', '日');  
console.log(list01);
```

// 配列の長さ

```
console.log(list01.length);    // 7
```

// 取り出し方

```
console.log(list01[0]);        // 月  
console.log(list01[2]);        // 水
```

配列はつよい

```
// 条件分岐なんていらなかったんや. . .

const hands = ['グー', 'チョキ', 'パー'];
const randomNumber = Math.floor(Math.random() * hands.length);

// 結果のテーブルをつくるのもありやな. . .
const result = [
  ['draw', 'win', 'lose'],
  ['lose', 'draw', 'win'],
  ['win', 'lose', 'draw'],
];
```


繰り返し処理 (for文)

繰り返し処理とは？？

```
// - 同じ処理を指定した回数実行するやり方
//   - 同じ処理を100回実行したいときに100回書くのか．．？
//   - 配列と相性が良い！

// - 書き方
for(開始条件； 繰り返しを続ける条件； 回数のカウンターの仕方){
    実際に繰り返したい処理    // {}内の処理が繰り返し実行される
}

// - 「i」の値を使って繰り返しをコントロールする．
// - 条件をセミコロン「;」で区切る点に注意！！
```

繰り返し処理とは？？

```
// 例
for(let i=0; i < 100; i++){    // 「i=0で実行」「i=1で実行」...の繰り返し
    console.log(i);           // {}内の処理が繰り返し実行される
}

// 上のコードは以下と同じ意味
console.log(0);    // 「let i=0」なので0からスタート
console.log(1);    // 「i++」なのでiは1ずつ増えて実行される
console.log(2);
// ...
console.log(98);
console.log(99);   // 「i<100」なので99まで実行したら終了
```

配列と組み合わせる

```
// 例
const array = ["G's", "ACADEMY", "FUKUOKA"];

for(let i=0; i < array.length; i++){
  console.log(array[i]); // arrayの0番目, 1番目, 2番目が順に出力される
}

// 上のコードは以下と同じ意味
console.log(array[0]);
console.log(array[1]);
console.log(array[2]); // 「i < array.length-1」なのでi=2で終了
```

【参考】for文の亜種(全部同じ結果)

```
const list = [1, 2, 3, 4, 5];

for (let i = 0; i < list.length; i++) {
  console.log(list[i]);
}

for (let x of list) {
  console.log(x);
}

list.forEach(function (x) {
  console.log(x);
})
```

オブジェクト (object)

オブジェクトとは？？

```
// オブジェクト
//   - 配列と同様に複数の値を管理する方法
//   - 配列の「インデックス」に対して「キー」「バリュー」で管理.
```

```
// 例
```

```
const class01 = {
  date: '2020/05/16',
  youbi: 'Sat',
  event: '第2回授業',
};
console.log(class01.event);           // 第2回授業
console.log(class01['event']);        // 第2回授業
```

- キー(名前)
date, youbi, event
- バリュー(値)
2020/05/16, Sat, 第2回授業

オブジェクトとは？？

- オブジェクト
 - オブジェクトの中身のとり方が大事！！
 - ひとまずは狙ったものを取れるようになればOK！

配列とオブジェクトの組み合わせ

// オブジェクトが複数あるときは配列と組み合わせる

// 例（オブジェクトの配列）

```
const class = [  
  { date: '2020/04/25', youbi: 'Sat', event: '第0回授業' },  
  { date: '2020/05/09', youbi: 'Sat', event: '第1回授業' },  
  { date: '2020/05/16', youbi: 'Sat', event: '第2回授業' },  
];
```

```
console.log(class[0].date); // '2020/04/25'  
console.log(class[1].event); // '第1回授業'
```

今後たくさん出てくる！！
この形の処理が大事！！

配列 & オブジェクトの使いどころ

API(外部サービス連携)でよく使う

(index):54

```
▼ {data: Array(5), status: 200, statusText: "OK", headers: {...}, config: {...}, ...} ⓘ  
  ► config: {adapter: f, transformRequest: {...}, transformResponse: {...}, timeout: 0, xsrfCookieName: "XSRF-T0...  
  ▼ data: Array(5)  
    ► 0: {...}  
    ► 1: {...}  
    ► 2: {...}  
    ► 3: {...}  
    ► 4: {...}  
    length: 5  
    ► __ob__: Observer {value: Array(5), dep: Dep, vmCount: 0}  
    ► __proto__: Array  
  ► headers: {content-type: "application/json; charset=utf-8"}  
  ► request: XMLHttpRequest {onreadystatechange: f, readyState: 4, timeout: 0, withCredentials: false, upload...  
    status: 200  
    statusText: "OK"  
  ► __proto__: Object
```

まずは触って慣れよう！

- 配列の練習
 - [list/array.html](#)
- オブジェクトの練習
 - [list/object.html](#)
- 上記が楽勝な人
 - [list/challenge.html](#)

localStorage (メモ帳)

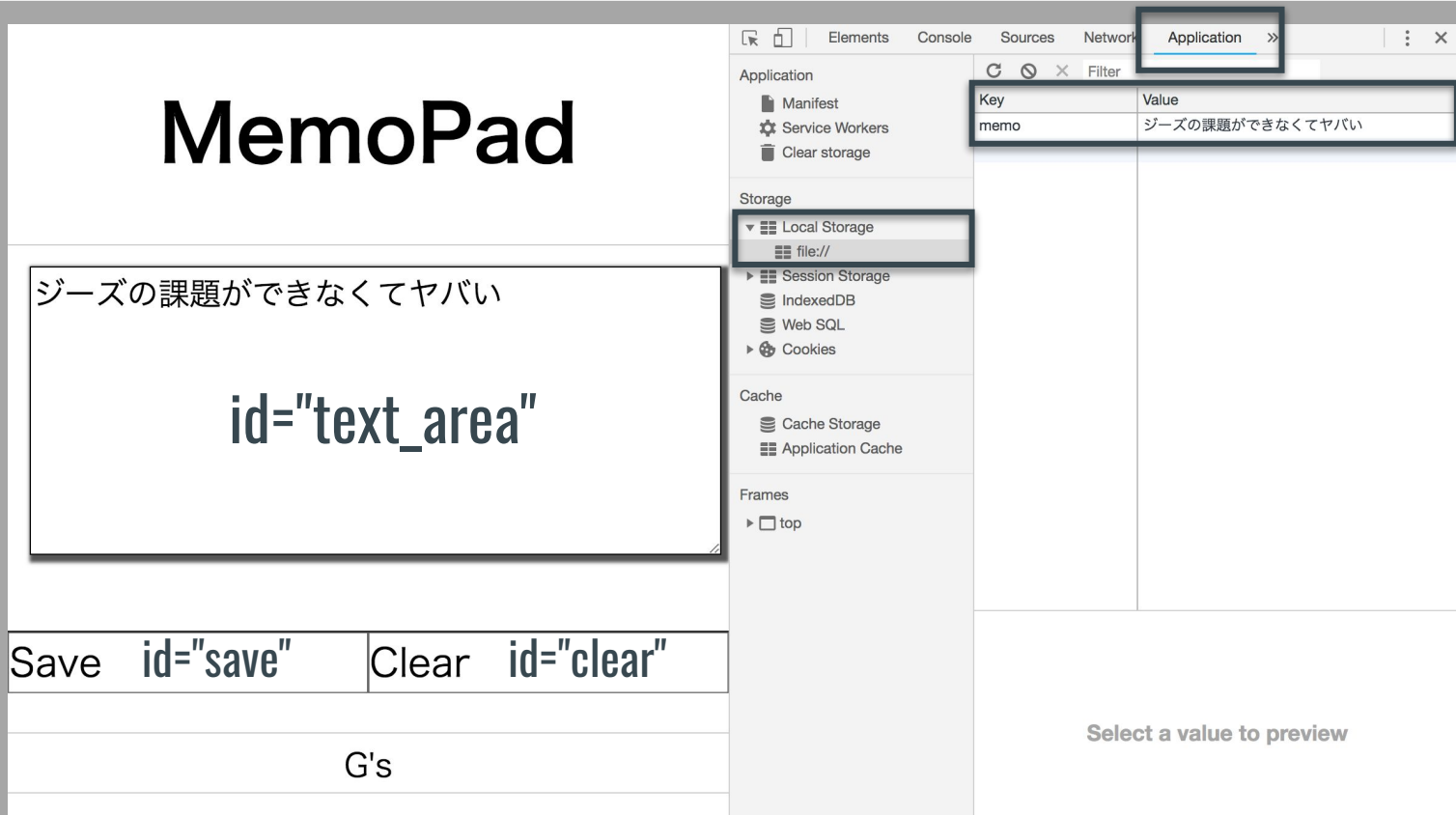
本日実装するアプリケーション

- memopad
 - ブラウザ上のテキストエリアにメモを入力.
 - ブラウザに情報を保存.
 - リロードしても情報が保存されるようにする.
- 仕様
 - saveボタンをクリックしたら情報を保存.
 - clearボタンをクリックしたら情報を削除.
 - リロード時に情報が保存されていれば自動的に表示.

「localStorage」を使う！

- 変数やオブジェクトの問題
 - ブラウザをリロードすると初期化される.
 - 次回読み込んだときまで値を保存したいときにつらい！
- localStorageができること
 - ブラウザにデータを保存できる！
 - 「オリジン単位」ごとに5MBまで保存できる.
 - （自分で削除しないと永続的に保存される）
- ※オリジン単位とは「http://www.example.com:80」のような
- 「プロトコル://ドメイン名:ポート番号」のこと.

完成イメージ



localStorageの基本

使い方

- 「key」と「value」を設定.
- keyは名前, valueは値.

key	value
memo	ジーズの課題ができなくてヤバい

localStorageの基本

内容	書き方
データ登録	<code>localStorage.setItem(key, value);</code>
データ取得	<code>localStorage.getItem(key);</code>
データ更新	<code>localStorage.setItem(key, value);</code>
データ削除	<code>localStorage.removeItem(key);</code>
データ全削除	<code>localStorage.clear();</code>
データの長さ	<code>localStorage.length;</code>
データの順番	<code>localStorage.key(インデックス);</code>

必要な処理

- ①データを保存するときの処理
 - saveボタンのクリックでイベント発火.
 - テキストエリアの情報を取得.
 - localStorageに情報を保存.
- ②データを削除するときの処理
 - clearボタンクリックでイベント発火.
 - localStorageの情報を消去.
 - 画面上の入力欄をクリア.

必要な処理

- ③データを取得するときの処理
 - 読み込み時にlocalstorageのデータ有無を確認.
 - 保存されていれば情報を取得.
 - テキストエリアに情報を表示.

書きながら覚えよう！！

// ①データを保存するときの処理

```
$('#save').on('click', function () {           // saveボタンクリックイベント
  const text = $('#text_area').val();           // .val()でtextarea内容を取得
  localStorage.setItem('memo', text);           // localStorageに保存
});
```

// Point: 「localStorage」は手打ちせず補完に頼る！！

書きながら覚えよう！！

// ②データを削除するときの処理

```
$('#clear').on('click', function () {  
    localStorage.removeItem('memo');  
    $('#text_area').val('');  
});
```

// clearボタンクリックイベント
// key名を指定して削除
// textareaを空文字で上書き

// Point: 「.val()」は()に何か入れると入れた値で上書きする！

書きながら覚えよう！！

// ③データを取得するときの処理

```
if (localStorage.getItem('memo')) { // 値が保存されていれば
  const text = localStorage.getItem('memo'); // データを取得
  $('#text_area').val(text); // 取得したデータで上書き
}
```

// Point: データが保存されているときのみ取得するよう条件分岐！

メモ帳を動かすところまで作ろう！
(memo01.html)

メモ帳中級編

1件しかデータを保存できない . . .

複数のデータをまとめたい！

JSON (データ形式の一種)

異なる言語下でデータをやり取りしたい... !

JavaScript

配列
オブジェクト

PHP
Python
localStorage

しかし、言語ごとにデータの形式は異なる...

JavaScript

オブジェクト

PHP
Python
localStorage

JSON(文字列の一種)なら幅広い言語で扱える！！

JavaScript

OK 👍

JSON形式のデータ

PHP
Python
localStorage

「タイトル」と「本文」の両方を保存してみよう！

MemoPad中級編

第2回課題

lolacstorageを使ってなにかつくろう！！

Save Clear

id="input"

id="text_area"

id="save" id="clear"

Application

Storage

Cache

Frames

manifest

Service Workers

Clear storage

Local Storage

file://

Session Storage

IndexedDB

Web SQL

Cookies

Cache Storage

Application Cache

top

Key

Value

memo02

{"title": "第2回課題", "text": "lolacstorage..."}

Select a value to preview

ヒント！！

```
// データをまとめて、JSON形式にして保存！

const data = {          // オブジェクト形式でタイトルと本文をまとめる
  title: 'test',
  text: 'hogehege'
};

const jsonData = JSON.stringify(data);    // dataをJSON形式に変換
localStorage.setItem('memo', jsonData);  // JSON形式で保存
```

ヒント！！

```
// JSON形式のデータをlocalStorageから取得して、オブジェクトに戻す.
```

```
const jsonData = localStorage.getItem('memo'); // データ取得  
const data = JSON.parse(jsonData);           // JSON形式から戻す  
console.log(data)                             // 表示
```

タイトル本文を保存してみよう！
(memo02.html)

課題

【課題】メモ帳アプリを実装！！

- localStorageを使った何かを作ろう
 - タブで複数記述できるメモ帳
 - じゃんけんのスコア保存
 - ゲームのセーブ&ロード機能
 - 保存したデータをグラフ表示

締切は次回授業前木曜「23:59:59」

やばいいい... (ゝ;ω;´)

完全に詰んだ．．．という方は

写経

※ 写経とは ※

誰かが書いた「動くコード」をひたすら書き写すこと

```
1  <!DOCTYPE html>
2  <html lang="ja">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>myMemoPad中級</title>
9      <link rel="stylesheet" href="css/sample.css">
10 </head>
11
12 <body>
13     <header>
14         <h1>MemoPad中級編</h1>
15     </header>
16
```

```
17     ....<main>
18     ....<input type="text" id="input">
19     ....<textarea id="text_area"></textarea>
20     ....<ul>
21     ....<li id="save">Save</li>
22     ....<li id="clear">Clear</li>
23     ....</ul>
24     ....</main>
25
26     ....<footer>
27     ....<small>G's</small>
28     ....</footer>
29     ....<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
30     ....<script>
31     ....$(function () {
32
```

```
33 ..... //1.Save・クリックイベント
34 ..... $('#save').on('click', function() {
35 .....     var title = $('#input').val();
36 .....     var text = $('#text_area').val();
37 .....     var obj = {
38 .....         "title": title,
39 .....         "text": text
40 .....     };
41 .....     var json = JSON.stringify(obj);
42 .....     localStorage.setItem('memo02', json);
43 ..... });
44
45
46 ..... //2.clear・クリックイベント
47 ..... $('#clear').on('click', function() {
48 .....     localStorage.removeItem('memo02');
49 .....     $('#input').val("");
50 .....     $('#text_area').val("");
51 ..... });
52
```

```
54 .....//3.ページ読み込み：保存データ取得表示
55 .....if (localStorage.getItem('memo02'))-{
56 .....    var json_get = localStorage.getItem('memo02');
57 .....    var obj_get = JSON.parse(json_get);
58 .....    $('#input').val(obj_get.title);
59 .....    $('#text_area').val(obj_get.text);
60 .....}
61 .....
62 .....});
63 .....
64 .....</script>
65 .....
66 </body>
67 .....
68 </html>
```

「写経」はあくまでヒント！
「オリジナルの作品」で力がつく！

P2Pタイム

まずはチーム内で解決を目指す！

訊かれた人は苦し紛れでも応える！！