



G's ACADEMY
TOKYO

Memo Pad



アジェンダ

❖ MemoPad

配列

繰り返し処理

localStorage

メモ帳

複数メモ（ヒント）

演習

配列

- Array -

JavaScriptの配列

【 配列 (array) 】

複数の値に順番をつけてまとめて扱う方法です。
配列の順番を識別する番号を「**インデックス**」と呼びます。
インデックスは「**0**」から始まります。

```
<script>  
  const list1 = ['大吉','中吉','小吉','吉','凶'];  
  const list2 = new Array('大吉','中吉','小吉','吉','凶');  
</script>
```

【 配列のアクセスイメージ (array) 】

インデックス「0」から値が**格納**されます。
list[1]では「css」が取得可能



配列を扱う関数

<https://qiita.com/takeharu/items/d75f96f81ff83680013f>

反復処理

JavaScriptの基礎

【反復処理 (Iterate)】

プログラム中で、ある条件が満たされているかどうかによって、次に実行するコードを切り替える命令

・ for文

条件が真の間だけ処理を続けていくループ文

・ while文

条件が真の間だけ処理を続けていくループ文

・ for in文, foreach文

配列/オブジェクトでループするとき（中の数がわからなくても使える）

【インクリメント・デクリメント (increment)】

演算子の短縮したようなものです。

i++	i += 1	i = i + 1	インクリメント
i--	i -= 1	i = i - 1	デクリメント

JavaScriptの基礎

【 for 処理 】

for文を使用することで、反復処理をおこなうことができる

```
for( 初期値; 条件式; 再初期値 ){  
    条件式の結果がtrueの場合実行されるスクリプト  
}
```

例) for文

```
for( let i=0; i<10; i++ ){ //条件 : iより1 0が大きい場合=true  
    console.log( i );      // 変数iに代入されてる値を表示  
}
```

練習 : for文

```
//1.変数の入れ物を作成  
let str="";  
//2.繰返し処理で、文字列を作成  
for ( let i=0; i<10; i++ ) {  
    str += "ループ：" + i + "回目<br>";  
}  
//3.変数「str」に入ってる文字列を pタグid="view"に表示  
$("#view").html( str );
```


【 テンプレートリテラル 】

従来の変数と文字列を連結させる方法は「+」演算子を使っていました。
ES6(javascript2015)からテンプレートリテラルを使用することでもっとシンプルに連結出来ます。

```
<script>
  const name1    = 'ジーズ';
  const name2    = '太郎';
  const str      = `こんにちは「 ${name1} ${name2} 」です`;
</script>
```

【 バッククオート： 入力方法 】

OS	入力キー
Win	「 shift 」 + 「 @ 」
Mac	「 shift 」 + 「 @ 」

JavaScriptの基礎

【 for 処理 】

for文を使用することで、反復処理をおこなうことができる

```
for( 条件式 ){  
    条件式の結果がtrueの場合実行される処理  
}
```

例) for文

```
for( let i=0; i<10; i++ ){  
    console.log( i ); // 変数iに代入  
}
```

練習 : for文

```
//1.変数の入れ物を作成  
let str="";  
//2.繰り返し処理で、文字列を作成  
for ( let i=0; i<10; i++ ) {  
    str += `ループ : ${i}回目<br>`;  
}  
//3.変数「str」に入ってる文字列を pタグid="view"に表示  
$("#view").html( str );
```

先程の文字作成がシンプルにできる！！

"ループ：" + i + "回目
";

`ループ : \${i}回目
`;

配列と反復処理の応用

【例）配列と反復処理】

配列と一緒に使用することが多い。

//1.配列を作成（必要な分だけ。。。今回は適当数に記述）

```
let ary = ["A","B","C","D","E","F","G","H","I","J"];
```

//2.変数の入れ物を作成

```
let str = "";
```

//3.繰返し処理で、文字列と配列を組み合わせ作成

```
for ( let i=0; i<10; i++ ) {  
    str += ` 配列 : ${ary[i]}<br> `;           //配列:ar[i]  
}
```

//4.変数「str」に入ってる文字列を pタグid="view"に表示

```
$("#view").html(str);
```

【例）反復処理】

HTML要素を作成する

<!-- ここにセレクトボックスの値が生成されます -->

<select id="date"></select>

<script>

//1.変数strを作成:空文字

let str = "";

//2.繰り返し処理で、HTMLタグと文字を作成

for(let i=2011; i<2099; i++) {

 str += `<option> \${ i } </option>`;

}

//4.変数「str」に入ってる文字列を pタグid="view"に表示

\$("#date").html(str);

</script>

localStorage

WebStorage

シンプルに使える localStorage

ブラウザ内に永続的にデータを保存するストレージ。

保存は「ドメイン名:ポート番号」の組み合わせ「オリジン」単位で保存されます。（例：<http://www.localhost:80>）

「オリジン」が同じであればブラウザを閉じた後も再度データにアクセス可能。

※ :80はブラウザが自動で付与してるの人間は入力していません。

保存量は「オリジン単位：10M」 保存期間は特になし。

自身の意志で削除しない限りデータは残ります。

シンプルに使える ～ localStorage ～

属性	説明
DATA取得	localStorage.getItem(KEYネーム);
DATA登録or更新	localStorage.setItem(KEYネーム, 値);
DATAを全削除	localStorage.clear();
1レコード削除	localStorage.removeItem(KEYネーム);
DATA数 (長さ)	localStorage.length
DATA取得 (何番目)	localStorage.key(インデックス)

実習！

MemoPad

id="text_area"

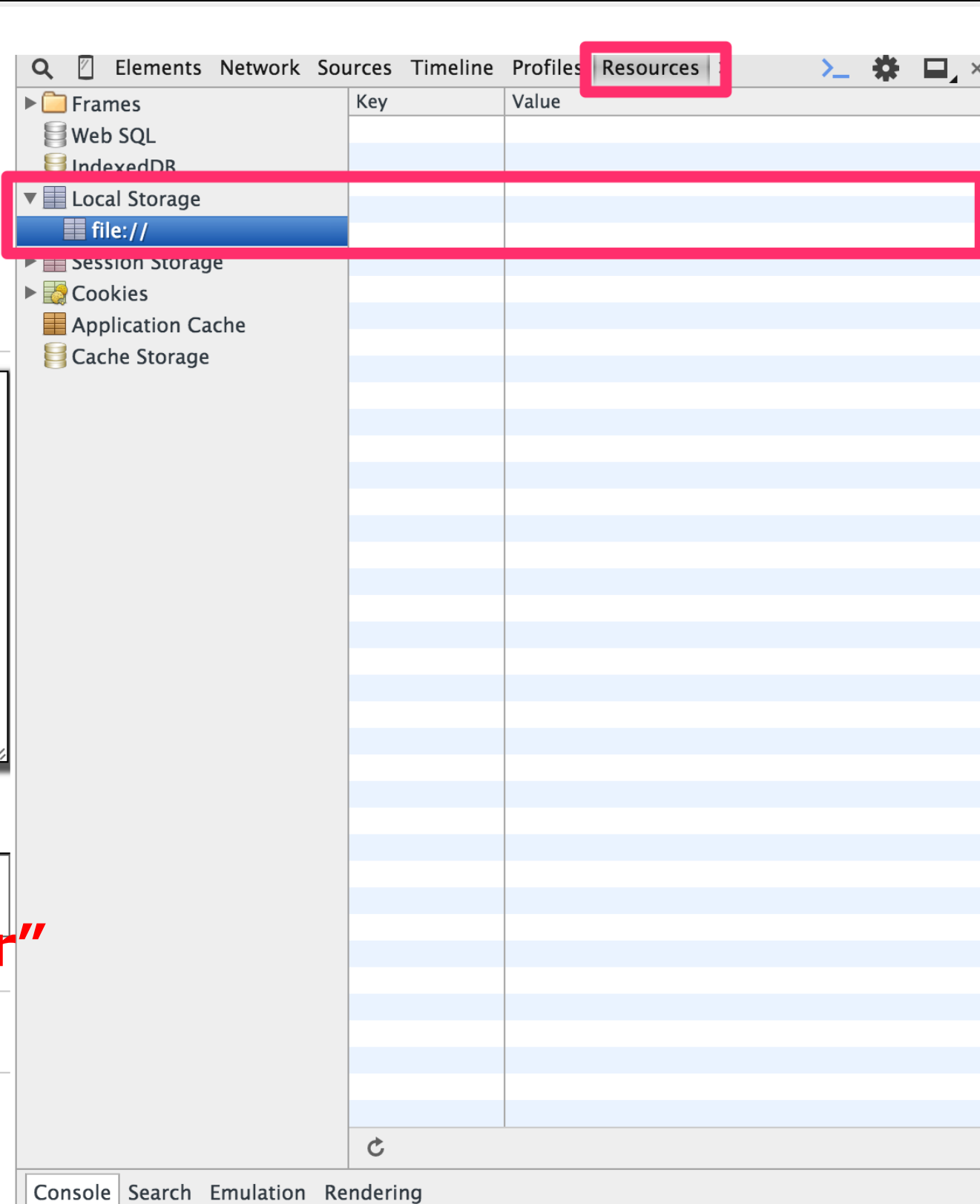
Save

id="save"

Clear

id="clear"

G's



HTML

```
<header>
  <h1>MemoPad</h1>
</header>

<main style="display:none;">
  <textarea id="text_area"></textarea>…入力表示
  <ul>
    <li id="save">Save</li>      . . . データ保存
    <li id="clear">Clear</li>    . . . データ削除
  </ul>
</main>

<footer><small>G's</small></footer>
```

JS

```
//1.Save クリックイベント  
$("#save").on("click",function(){  
    const value = $("#text_area").val();  
    localStorage.setItem("memo", value);  
    alert("Saveしました");  
});
```

JS

```
//2.clear クリックイベント  
$("#clear").on("click",function(){  
    localStorage.removeItem("memo");  
    alert("Clearしました");  
    $("#text_area").val("");  
});
```

JS

//3.ページ読み込み：保存データ取得表示

```
if(localStorage.getItem("memo")){  
    const value = localStorage.getItem("memo");  
    $("#text_area").val(value);  
}
```

JS

//ロード時に表示

//※こんなこともできる例

```
$("#main").slideDown(2000)
```

```
.animate({top:100,left:100}, 1000)
```

```
.animate({top:-50,left:-100}, 1000)
```

```
.animate({top:0,left:0}, 500);
```

記述式 : `$("#***").animate({移動座標}, ミリ秒); //1000= 1 秒`

複数メモ

kadai/index2.html

複数登録の練習サンプル

※答えではありません。答えは自分で作り上げます。

//登録処理

```
localStorage.setItem("memo1","リンゴ");
```

```
localStorage.setItem("memo2","ブドウ");
```

```
localStorage.setItem("memo3","レモン");
```

//Key名不明でも取得可能な処理

```
for(let i=0; i<localStorage.length; i++){
```

```
    const k = localStorage.key( i ); //key名取得
```

```
    console.log( k ); //key名表示
```

```
    console.log( localStorage.getItem( k ) ); //value取得表示
```

```
}
```

課題

【課題】 MemoPadアプリを再作成

◇ 課題仕様

他なんでもあり！！localStorage使ってれば。

- ・じゃんけんの点数を記憶させておく（履歴を取るなど）
- ・付箋アプリ。EverNote...とか

◇ プラス α (難しい！今の自分に合わせて使いましょう)

「複数メモ登録」ができるようになるGood!

※forを使って、複数表示を可能しましょう！

※削除機能もあるといいですね。

<http://www.webopixel.net/javascript/91.html>