

## 練習問題 7

作成ファイル：html ファイル「practice7.html」、script ファイル「script7.js」

課題：読み込み時、以下の script に処理を追加し、配列内のデータを「実行結果」のようにコンソールに出力せよ。

ヒント：配列を繰り返したいときは以下の構文を使用する（java でいう拡張 for 文）  
each ブロックの {} 文の中の処理は、配列の要素数だけ繰り返されます

```
$.each(配列名, function (index, value) {  
    // ↑function 引き数の 1 つ目は要素番号、2 つ目に配列の中身(データ)が入る  
    // ここに繰り返したい処理を記述  
  
    console.log(index);    // 例: インデックス番号をコンソールに表示  
    console.log(value);    // 例: 中身をコンソールに表示  
  
});  
  
// ちなみに、Java のように for 文で初期値と繰り返しの条件式を指定する方法でも書けるが、  
// 全件表示等では使用しないことが多い  
for (var i = 0; i < 配列名.length; i++) {  
    // ここに繰り返したい処理を記述  
  
    console.log(i);        // 例: インデックス番号をコンソールに表示  
    console.log(配列名[i]); // 例: 中身をコンソールに表示  
  
}
```

### 【script7.js】

```
$(function () {  
    var menu = ["牛とろ玉うどん", "釜玉うどん", "おろし醤油うどん", "ぶっかけうどん"];  
    // ここに処理を追加する  
  
});
```

### 【practice7.html】

```
<!DOCTYPE html>  
<html lang="ja">  
  
<head>  
    <meta charset="UTF-8">  
    <title>練習問題 7</title>  
</head>  
  
<body>
```

```

    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="script7.js"></script>
</body>

</html>

```

実行結果：



## 練習問題 8

作成ファイル：html ファイル「practice8.html」、script ファイル「script8.js」

課題：読み込み時、以下の script に処理を追加し、配列内のデータを「実行結果」のように画面に出力せよ。

ヒント：最終的に以下のような HTML ができあがる

```

<ul id="menu-list">
    <!--ここにメニューを追加-->
    <li>1. 牛とろ玉うどん</li>
    <li>2. 釜玉うどん</li>
    <li>3. おろし醤油うどん</li>
    <li>4. ぶっかけうどん</li>
</ul>

```

「practice8.html」

```

<!DOCTYPE html>
<html lang="ja">

<head>
    <meta charset="UTF-8">
    <title>練習問題 8</title>
</head>

<body>
    <ul id="menu-list">

```

```

    <!--ここにメニューを追加-->
  </ul>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="script8.js"></script>
</body>

</html>

```

「実行結果」

- 1. 牛とろ玉うどん
- 2. 釜玉うどん
- 3. おろし醤油うどん
- 4. ぶっかけうどん

## 練習問題 9

作成ファイル：html ファイル「practice9.html」、script ファイル「script9.js」

課題： 「practice9.html」 のリスト内の要素を取得し、テキストの語尾に「.java」を追加せよ

ヒント：配列と同じように、画面上の要素に対しても繰り返し構文が使用できる

```

$.each($(要素), function (index, element) {
    //繰り返したい処理を記述
    console.log($(element).text()); //例:要素ごとのテキストデータをコンソールに表示
});

```

「practice9.html」

```

<!DOCTYPE html>
<html lang="ja">

<head>
  <meta charset="UTF-8">
  <title>練習問題 9</title>
</head>

<body>
  <p>提出物</p>
  <ul id="java-file">

```

```

<li>Bmi</li>
<li>FizzBuzz</li>
<li>Leap</li>
<li>Multiplication</li>
<li>Employee</li>
<li>StringToInt</li>
<li>Bird</li>
<li>ArgsSort</li>
<li>Janken</li>
</ul>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script src="script9.js"></script>
</body>
</html>

```

「実行結果」

---

## 提出物

- Bmi.java
- FizzBuzz.java
- Leap.java
- Multiplication.java
- Employee.java
- StringToInt.java
- Bird.java
- ArgsSort.java
- Janken.java

## 練習問題 10

作成ファイル：html ファイル「practice10.html」、script ファイル「script10.js」

課題：「script10.js」に処理を追加し、連想配列の情報を li タグで「実行結果」のように画面に出力せよ。

ヒント：連想配列(Java でいう HashMap)に対しても繰り返し構文が使用できる

```

var sampleData = { "みかん": "炬燵で食べたい果物 NO1", "すいか": "そろそろ季節ごろ" }
$.each(sampleData, function (key, value) {
    console.log('キーは' + key + ', バリューは' + value);
});

```

```
});
```

each{}文の中の処理は、sampleData の要素数だけ繰り返されます

その際、sampleData のキー(プロパティ)が引数の 1 番目に入り、バリュー(データ)が引数の 2 番目に入ります

↓イメージ

```
var sampleData = { "みかん": "炬燵で食べたい果物 NO1", "すいか": "そろそろ季節ごろ" }  
$.each(sampleData, function (key, value) {  
    console.log('キーは' + key + ', バリューは' + value);  
});
```

「script10.js」

```
$(function () {  
    var profile = { name: "bonobono", age: 4, type: "ラッコ" };  
  
    //ここに処理を追加  
  
});
```

「practice10.html」

```
<!DOCTYPE html>  
<html lang="ja">  
  
<head>  
    <meta charset="UTF-8">  
    <title>練習問題 10</title>  
</head>  
  
<body>  
    <h3>ぼのぼののプロフィール</h3>  
    <ul id="bono-profile">  
  
    </ul>  
    <script type="text/javascript"  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>  
    <script src="script10.js"></script>  
</body>  
  
</html>
```

「実行結果」

## ぼのぼののプロフィール

- 名前は、ぼのぼの
- 歳は、4
- 種別は、ラッコ

### 演習問題 11

作成ファイル: html ファイル「practice11.html」、script ファイル「script11.js」、css ファイル「practice11.css」

課題: 「script11.js」に処理を追加し、「実行結果」のように画面に出力せよ。

「script11.js」

```
$(function () {  
    var seaCreatures = [  
        {  
            "名前": "シロギス",  
            "学名": "Sillago japonica",  
            "説明": "底生魚で、群れを作り外敵から逃れる。摂餌のために砂に潜る。砂浜からの投げ釣り、沖釣りで人気が高い。"  
        },  
        {  
            "名前": "アオリイカ",  
            "学名": "Sepioteuthis lessoniana",  
            "説明": "外套背長 20 センチ前後。外套膜(身体の周り)を「えんぺら」と呼ぶ鰭が縁取っている。餌木(エギ)と呼ばれる、日本の伝統的なルアーを使った釣りが人気。"  
        },  
        {  
            "名前": "メジナ",  
            "学名": "Girella punctata",  
            "説明": "体型は扁平な卵型で、灰色がかった青緑色。クロダイと双壁をなす磯釣りの人気魚種。寄せエサを用いたウキフカセ釣りでねらう。"  
        }  
    ]  
})
```

```
//ここに処理を追加する

});
```

「practice11.html」

```
<!DOCTYPE html>
<html lang="ja">

<head>

  <meta charset="UTF-8">
  <title>練習問題 11</title>
  <link rel="stylesheet" type="text/css" href="practice11.css">
</head>

<body>
  <section id="fish_guide">
    <h1>釣りもの事典</h1>
    <!-- ここに追加 -->
  </section>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="script11.js"></script>
</body>

</html>
```

「practice11.css」

```
.target{
  margin: 20px;
  padding: 10px;
  background-color: rgba(229, 255, 158, 1);
}
```

## 「実行結果」

### 魚事典

名前： シロギス

学名： *Sillago japonica*

説明： 底生魚で、群れを作り外敵から逃れる。摂餌のために砂に潜る。砂浜からの投げ釣り、沖釣りで人気が高い。

名前： アオリイカ

学名： *Sepioteuthis lessoniana*

説明： 外套背長20センチ前後。外套膜（身体の周り）を「えんぺら」と呼ぶ鱗が縁取っている。餌木（エギ）と呼ばれる、日本の伝統的なルアーを使った釣りが人気。

名前： メジナ

学名： *Girella punctata*

説明： 体型は扁平な卵型で、灰色がかった青緑色。クロダイと双璧をなす磯釣りの人気魚種。寄せエサを用いたウキフカセ釣りでねらう。