

SVG 資料第 10 回目 (その 2)

Ajax による同期通信と非同期通信の違い

メディア専門ユニット I(SVG)

2017/6/27

あるプログラムから別のプログラムにデータを送る場合を考える。

- ▶ データを送った先のプログラムから応答を待つとその先の処理を行うのが同期通信
- ▶ 応答を待たずに、処理を続けるのが非同期通信
- ▶ 非同期通信では応答が返るまでに別の処理ができる。

素数の数を数える

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

ここでは 100 万以下の素数を数えて表示するページを考える。

- ▶ 10 万ごとの区間に区切り、その間の素数を数える。
- ▶ それぞれの区間ごとにサーバーに計算を要求
- ▶ 同期通信と非同期通信の選択が可能

素数の数を数える-HTML ファイル

第 10 回目 (その
2)

メディア専門ユニッ
ト I (SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
1<!DOCTYPE html>
2<html>
3  <head>
4    <meta charset="utf-8"/>
5    <script type="text/ecmascript" src="make-svg-elm.js"></script>
6    <script type="text/ecmascript" src="countPrimes-Ajax.js"></script>
7    <style type="text/css">
8      td { text-align:right;}
9    </style>
10   <title>素数の数を数える</title>
11 </head>
12 <body>
13   <h1 class="head">素数の数を数える</h1>
14   <div>
15     <form name="Test" method="PUT" action="javascript:getData()">
16       <div><input type="radio" name="ajax-mode" value="true">非同期
17         <input type="radio" name="ajax-mode" value="false">同期</div>
18       <div><input type="SUBMIT" value="実行"/></div>
19     </form>
20   </div>
21   <div>
22     <table></table>
23   </div>
24 </body>
25</html>
```

素数の数を数える-HTML ファイル (解説)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

- ▶ 5 行目で HTML 要素を作成するライブラリーを読み込む
- ▶ 6 行目で Ajax の処理をするファイルを読み込む
- ▶ 7 行目から 9 行目は table 内の td 要素の表示を右寄せにするスタイルの指定
- ▶ 16 行目から 17 行目は同期通信と非同期通信を選択するラジオボタン。属性 name が ajax-mode
- ▶ 18 行目は計算を実行するボタン
- ▶ 22 行目に結果を表示するための table 要素 (中身がない)

素数の数を数える-JavaScript ファイル (1)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
1window.onload = function(){
2    window.Test["ajax-mode"].value = "true";
3}
4function createXMLHttpRequest( func ) {
5    let xmlHttpObj = new XMLHttpRequest();
6    if(xmlHttpObj) xmlHttpObj.onreadystatechange = func;
7    return xmlHttpObj;
8}
```

- ▶ 1 行目から 3 行目はファイルがロード終了後の処理。
 - ▶ ラジオボタンのうち非同期通信を選択
 - ▶ form 要素の属性 name が Test なので、この form 要素は window.Test で参照可能
 - ▶ したがってラジオボタンの値を window.Text["ajax-mode"].value で指定できる
- ▶ 4 行目から 8 行目は Ajax 通信のオブジェクトを作成する関数の定義 (既出)

素数の数を数える-JavaScript ファイル (2) 解説

Ajax 通信をするための関数 getData

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
9function getData() {  
10  let table = document.getElementsByTagName("table")[0];  
11  let P = table.parentNode;  
12  P.removeChild(table);  
13  table = MKHTMLelm(P, "table",{}, {});  
14  let request = 0, start = new Date();
```

- ▶ 10 行目で table 要素を得ている。
- ▶ 11 行目でその Ajax 通信をするための関数 getData を得て、10 行目で得た table 要素を DOM から取り除く。
- ▶ 13 行目で新たに table 要素を作成し、その要素の子要素にする (table 要素内の初期化)
- ▶ 14 行目で通信の状況をチェックするための変数を初期化
 - ▶ request はサーバーに出した要求の数
 - ▶ start はボタンが押されたときの時間を保存
単位はある時期からの経過時間で単位はミリ秒
(Processing の mils と同じ)

素数の数を数える-JavaScript ファイル (3)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
15 for(let i=0; i<10; i++) {
16     let httpObj = createXMLHTTPReq(function(){
17         if(httpObj.readyState == 4 && httpObj.status == 200) {
18             let tr = MKHTMLElm(table,"tr",{},{})
19             MKHTMLElm(tr, "td", {}, {}).
20                 appendChild(document.createTextNode(i*100000+1));
21             MKHTMLElm(tr, "td", {}, {}).
22                 appendChild(document.createTextNode(httpObj.responseText));
23             request--;
24             console.log(httpObj.responseText);
25             console.log(new Date()-start);
26             if(request <= 0) console.log("done");
27         }
28     });
29     if(httpObj) {
30         request++;
31         httpObj.open("GET", './countPrimes.php?N=${i*100000+1}',
32                     window.Test["ajax-mode"].value == "true");
33         httpObj.send(null);
34     }
35 }
36 }
```


素数の数を数える-JavaScript ファイル (解説)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

- ▶ 15 行目から 10 万の区間ごとに分けた素数の数をサーバーに要求を 10 個生成
 - ▶ 17 行目から 28 行目でリクエストを生成
 - ▶ 18 行目から 28 行目でリクエスト結果の処理
 - ▶ 19 行目から 20 行目と 21 行目から 22 行目で区間の開始位置とその区間に対する結果を表示する td 要素を作成
 - ▶ 23 行目で発行した要求の数を減らす
 - ▶ 24 行目で「実行」ボタンを押してからの経過時間をコンソールに表示
 - ▶ 26 行目でその値が 0 のときはメッセージを出力
- ▶ 29 行目から 35 行目で Ajax のオブジェクトが生成できたら、要求をサーバーに送る。
- ▶ 31 行目で要求の数のカウンタを増加
- ▶ 32 行目で非同期通信かどうかの判定した結果をそのままメソッドに渡す

素数の数を数える-PHP ファイル (1)

素数であるか同課の判定する関数

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
1<?php
2function isPrime($n, &$Primes, $No) {
3    for($j=0;$j<$No;$j++) {
4        $p = $Primes[$j];
5        if($n % $p == 0) return false;
6        if($n < $p*$p+$p) break;
7    }
8    return true;
9}
```

- ▶ 引数は、素数かどうか判定したい数、素数のリスト、素数のリストに含まれる素数の数の3つ
- ▶ 2番目の引数の前の&は参照渡しを示す。
- ▶ 与えられた数の平方根以下の素数で割り切られないならば素数と判定 (5行目と6行目)

素数の数を数える-PHP ファイル (2)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
10$start = (array_key_exists('N', $_GET))?$_GET['N']:$argv[1];
11$step  = 10*10000;
12$limit = 10000;
13$primes = [2];
14for($i=3;$i<$limit; $i+=2) {
15  if(isPrime($i, $primes, count($primes))) array_push($primes, $i);
16}
```

- ▶ 変数\$argv はコマンドプロンプトから実行したときの、引数のリストを表す配列
 - ▶ \$argv[0] は実行するファイル名
 - ▶ 2 番目以降はそれぞれ文字列が入る
- ▶ ここではサーバーにコピーする前にコマンドプロンプトからデバッグできるようにするために利用
- ▶ 関数 array_key_exists() は配列内にキーがあるかどうかを判定
- ▶ \$_GET 内に指定したキーがなければコマンドプロンプトからの実行と判定
- ▶ 11 行目で求める素数の区間の幅、12 行目で配列に格納する素数の大きさの上限を設定。
- ▶ 13 行目は素数の配列の初期化
- ▶ 14 行目から 16 行目まで指定された範囲の素数のリストを作成
- ▶ 関数 array_push() は指定した値 (2 番目以降の引数) を指定した配列 (1 番目の引数) の後ろに追加

素数の数を数える-PHP ファイル (2)

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

```
17$c = 0;
18if($start < $limit ) {
19  $c = count($primes);
20  $start = $limit;
21}
22$start = $start - $start % 2 + 1;
23$pNo = count($primes);
24$L = ($start < $step)?$step: ($start + $step);
25for($i=$start;$i<$L; $i+=2) {
26  if(isPrime($i, $primes, $pNo)) $c++;
27}
28print $c;
29?>
```

素数の数を数える-PHP ファイル (2) 解説

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

- ▶ 17 行目は与えられた区間内の素数の数を数えるカウンターの初期化
- ▶ 18 行目から 21 行目は保存してある素数より小さい区間で求める時の処理 (かなり手抜き)
- ▶ 22 行目は開始の値を奇数にするための処理
- ▶ 23 行目は素数のリストの大きさを保存
- ▶ 24 行目は区間の最後の数を設定
- ▶ 25 行目から 27 行目で与えられた区間の数に対して素数かどうかの判定をし、素数の数をカウント
- ▶ 28 行目で素数の数を出力

確認して報告

第 10 回目 (その
2)

メディア専門ユニッ
ト I(SVG)

同期通信と非同期
通信

Ajax の例

やってみよう

- ▶ 同期通信と非同期通信のときのページの結果がどうなるか
- ▶ コンソールを開いて出てくる値を調べる