[演習]素数の列挙

・素数を格納する空配列 primes を用意する
・for(let i=2; i<=100; i++){
　・iが素数なら配列 primes に i をpush
}

- 先のプログラムの body部 を書き換えて、100以下の素数を列挙するプログラムを作成しよう。

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_11-2</title>
    <script>
        function isPrime(n){
            if(n == 2){
                return true;
            } else if(n > 2){
                for(let i=2; i<n; i++){
                    if(n%i == 0){
                        return false;
                    }
                }
                return true;
            }
        }
        console.log("関数 isPrime のテスト");
        console.log(isPrime(7));
        console.log(isPrime(12));
    </script>
</head>
```

↑ここはさっきと同じ

```
<body>
    <p>
        <script>
            考えてみよう
        </script>
    </p>
</body>

</html>
```

素数を判定する関数 isPrime は
実際にはもっと効率化できる。
余裕のある人は考えてみよう。

100以下の素数は 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97 です。

[演習]素数の列挙（解答）

解答例

```
24
25  <body>
26      <p>
27          <script>
28              let primes = [];
29              for(let i=2; i<=100; i++){
30                  if(isPrime(i)){
31                      primes.push(i);
32                  }
33              }
34              document.write("100以下の素数は " + primes + " です。");
35          </script>
36      </p>
37  </body>
38
39  </html>
```

isPrime(i) == true
でもok

[演習] 分散を求める

$$s^2 = \overline{x^2} - (\overline{x})^2$$

- 分散を計算するために「配列の要素の平均を求める関数」「配列のすべての要素を2乗する関数」を作り、与えられた配列の要素の分散を計算するプログラムを作ろう。

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_11-3</title>
    <script>
        /*
        配列の平均を返す関数 calculateMean
        入力：配列
        出力：数値
        */
        function calculateMean(array){
            考えてみよう
        }

        /*
        配列のすべての要素を2乗する関数 squareElements
        入力：配列
        出力：配列
        */
        function squareElements(array){
            考えてみよう
        }
    </script>
</head>
```

```
<body>
    <p>
        <script>
            let scores = [65, 81, 73, 52, 84];
            考えてみよう
        </script>
    </p>
</body>

</html>
```

分散は 134 です。

※コメント部（緑色の字）は書き写さなくてよい。

[演習] 分散を求める（解答）

$$s^2 = \overline{x^2} - (\overline{x})^2$$

解答例

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_11-3</title>
    <script>
        function calculateMean(array){
            let result = 0;
            for(let i=0; i<array.length; i++){
                result += array[i];
            }
            result = result/array.length;
            return result;
        }

        function squareElements(array){
            let result = [];
            for(let i=0; i<array.length; i++){
                result.push(array[i]**2);
            }
            return result;
        }
    </script>
</head>

<body>
    <p>
        <script>
            let scores = [65, 81, 73, 52, 84];
            let variance = calculateMean(squareElements(scores)) - calculateMean(scores)**2;
            document.write("分散は " + variance + " です。");
        </script>
    </p>
</body>

</html>
```

[演習] a 以上 b 以下の整数をランダムに返す

- 整数 a, b に対して、a以上b以下の整数をランダムに返す関数 randomInteger を定義してみよう。
- 次のコードを参考にすること。

```
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_11-a</title>
7       <script>
8           /*
9           a 以上 b 以下の整数をランダムに返す関数 randomInteger
10          入力：数値 a,b （a < b を想定してよい）
11          出力：数値
12          */
13          function randomInteger(a,b){
14              return
15          }
16      </script>
17  </head>
18
19  <body>
20      <p>
21          <script>
22              let a = 10;
23              let b = 19;
24              document.write("関数 randomInteger(", a, ",", b, ") を使って指定した数を 10 個生成します。<br>");
25              for(let i=1; i<=10; i++){
26                  document.write(randomInteger(a,b), ", ");
27              }
28          </script>
29      </p>
30  </body>
```

考えてみよう

関数 randomInteger(10,19) を使って指定した数を 10 個生成します。
11, 19, 11, 15, 16, 10, 14, 18, 17, 14,

[演習] a以上b以下の整数をランダムに返す（解答）

解答例

```
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_11-a</title>
7       <script>
8           /*
9           a 以上 b 以下の整数をランダムに返す関数 randomInteger
10          入力：数値 a,b  (a < b を想定してよい)
11          出力：数値
12          */
13          function randomInteger(a,b){
14              return Math.floor((b-a+1)*Math.random() + a);
15          }
16      </script>
17  </head>
18
19  <body>
20      <p>
21          <script>
22              let a = 10;
23              let b = 19;
24              document.write("関数 randomInteger(", a, ",", b, ") を使って指定した数を 10 個生成します。<br>");
25              for(let i=1; i<=10; i++){
26                  document.write(randomInteger(a,b), ", ");
27              }
28          </script>
29      </p>
30  </body>
```

[演習]剰余算を正しく計算する関数

- 第5回の授業でも注意した通り、JavaScriptでは被除数が負の数の剰余は負の数で返す仕様になっている。
- これを改善し、被除数が負の数の剰余も正の数で返す関数 positiveMod を定義しよう。

```
5 % 3;      // 2
(-5) % 3; // -2 ← 本当は 1
6 % 3;      // 0
(-6) % 3; // -0 ← 本当は 0
```

```
<head>
    <meta charset="UTF-8">
    <title>Prog_12-b</title>
    <script>
        /*
        被除数が負の数でも正しく剰余算を計算する関数 positiveMod
        入力：数値 a,b （b は正の数を想定してよい）
        出力：数値
        */
        function positiveMod(a,b){
            考えてみよう
        }
    </script>
</head>

<body>
    <p>
        <script>
            let a = 5;
            document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
            a = -5;
            document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
            a = 6;
            document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
            a = -6;
            document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
        </script>
    </p>
</body>
```

5 % 3 の計算結果は 2 です。
-5 % 3 の計算結果は 1 です。
6 % 3 の計算結果は 0 です。
-6 % 3 の計算結果は 0 です。

[演習]剰余算を正しく計算する関数（解答）

解答例

```
4   <head>
5       <meta charset="UTF-8">
6       <title>Prog_12-b</title>
7       <script>
8           /*
9           被除数が負の数でも正しく剰余算を計算する関数 positiveMod
10          入力：数値 a,b （b は正の数を想定してよい）
11          出力：数値
12          */
13          function positiveMod(a,b){
14              if(a%b == -0){
15                  return 0;
16              } else if(a >= 0){
17                  return a%b;
18              } else{
19                  return a%b + b;
20              }
21          }
22      </script>
23  </head>
24
25  <body>
26      <p>
27          <script>
28              let a = 5;
29              document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
30              a = -5;
31              document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
32              a = 6;
33              document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
34              a = -6;
35              document.write(a, " % 3 の計算結果は ", positiveMod(a,3), " です。<br>");
36          </script>
37      </p>
38  </body>
```

b も 負の数を許すなら、
どのように修正すべきか？

別解

```
function positiveMod(a,b){
    return (a%b + b)%b;
}
```

[演習]ルートの中身を簡約化する

結構難しい NEW

- √12 = 2√3 のように、与えられた自然数 n に対して、正の平方根 √n を a√b と簡約化するときの a, b の値を求める関数 simplifyRoot を定義しよう。

```
<head>
    <meta charset="UTF-8">
    <title>Prog_12-c</title>
    <script>
        /*
        ルートの中身を簡約化する関数 simplifyRoot
        入力：数値 n （n は2以上の整数を想定してよい）
        出力：配列 [a,b] (a,b は √n = a√b と簡約化したときの a,b)
        */
        function simplifyRoot(n){

            考えてみよう

            return [a,b];
        }
    </script>
</head>

<body>
    <p>
        <script>
            for(let i=2; i<=20; i++){
                let a = simplifyRoot(i)[0];
                let b = simplifyRoot(i)[1];
                if(a == 1){
                    document.write("√", i, " はこれ以上整理できません。<br>");
                } else if(b == 1){
                    document.write("√", i, " を整理すると ", a, " です。<br>");
                } else{
                    document.write("√", i, " を整理すると ", a, "√", b, " です。<br>");
                }
            }
        </script>
    </p>
</body>
```

最初は a,b の値だけ表示させてもよい

√2 はこれ以上整理できません。
√3 はこれ以上整理できません。
√4 を整理すると 2 です。
√5 はこれ以上整理できません。
√6 はこれ以上整理できません。
√7 はこれ以上整理できません。
√8 を整理すると 2√2 です。
√9 を整理すると 3 です。
√10 はこれ以上整理できません。
√11 はこれ以上整理できません。
√12 を整理すると 2√3 です。
√13 はこれ以上整理できません。
√14 はこれ以上整理できません。
√15 はこれ以上整理できません。
√16 を整理すると 4 です。
√17 はこれ以上整理できません。
√18 を整理すると 3√2 です。
√19 はこれ以上整理できません。
√20 を整理すると 2√5 です。

[演習]ルートの中身を簡約化する（解答）

解答例

```
<head>
    <meta charset="UTF-8">
    <title>Prog_12-c</title>
    <script>
        /*
        ルートの中身を簡約化する関数 simplifyRoot
        入力：数値 n （n は2以上の整数を想定してよい）
        出力：配列 [a,b] （a,b は √n = a√b と簡約化したときの a,b）
        */
        function simplifyRoot(n){
            let a = 1;
            let b = n;
            for(let i=2; i <= Math.sqrt(n); i++){
                while(true){
                    if(b%(i*i) == 0){
                        a *= i;
                        b /= (i*i);
                    } else{
                        break;
                    }
                }
            }
            return [a,b];
        }
    </script>
</head>

<body>
    <p>
        <script>
            for(let i=2; i<=20; i++){
                let a = simplifyRoot(i)[0];
                let b = simplifyRoot(i)[1];
                if(a == 1){
                    document.write("√", i, "はこれ以上整理できません。<br>");
                } else if(b == 1){
                    document.write("√", i, "を整理すると", a, "です。<br>");
                } else{
                    document.write("√", i, "を整理すると", a, "√", b, "です。<br>");
                }
            }
        </script>
    </p>
</body>
```