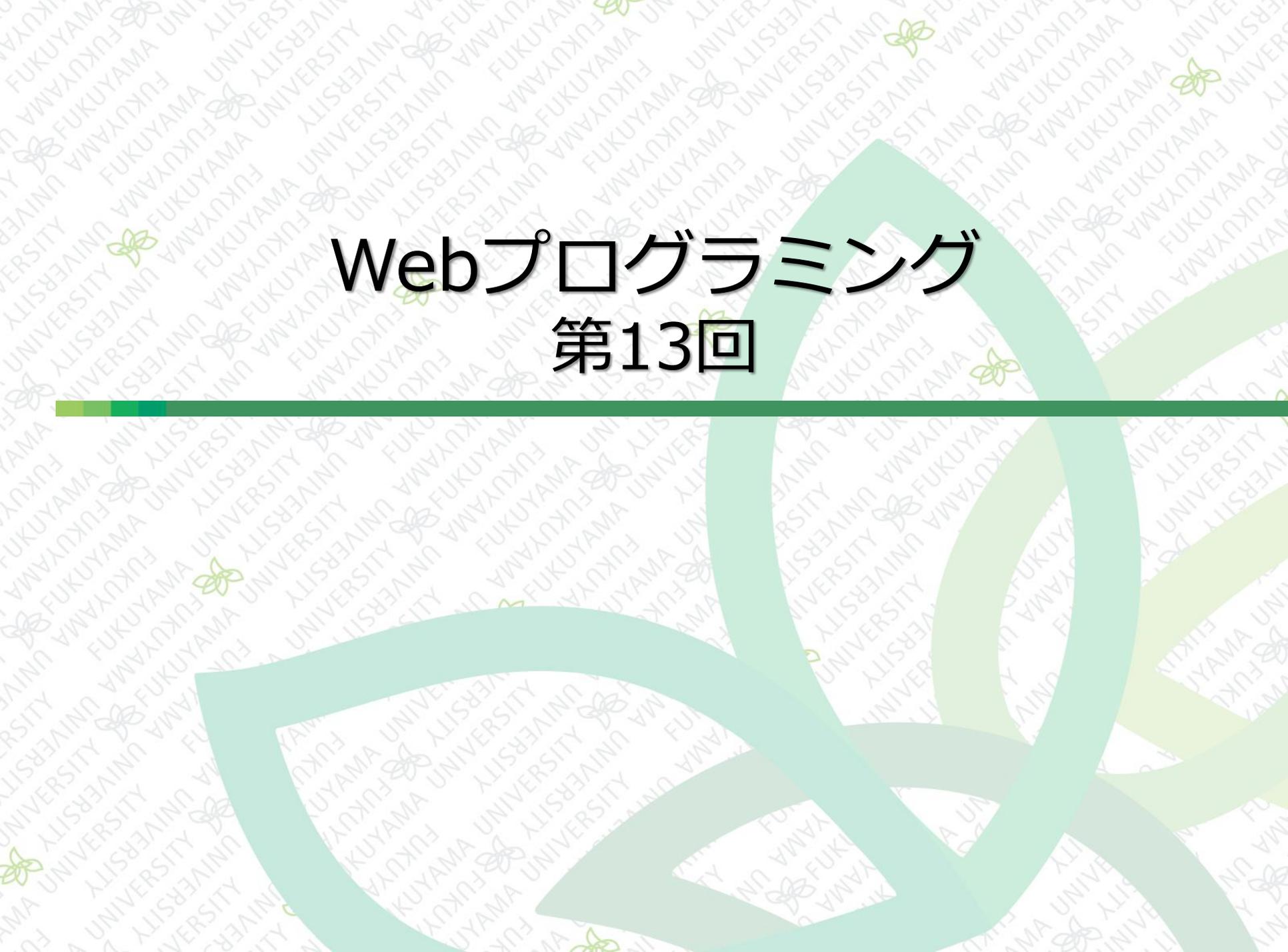


Webプログラミング 第13回



前回まで

■ オブジェクト

- 複数のプロパティを持つデータ

- for ... inを用いて表示

- HTMLに出力

■ フォームの入力とイベント

- イベント

- フォーム、テキストフィールド

- HTMLに出力

今回の内容

■ 日付と時間の処理

- 年月日と日付をわかりやすく表示する

- Dateオブジェクト

- 日数計算

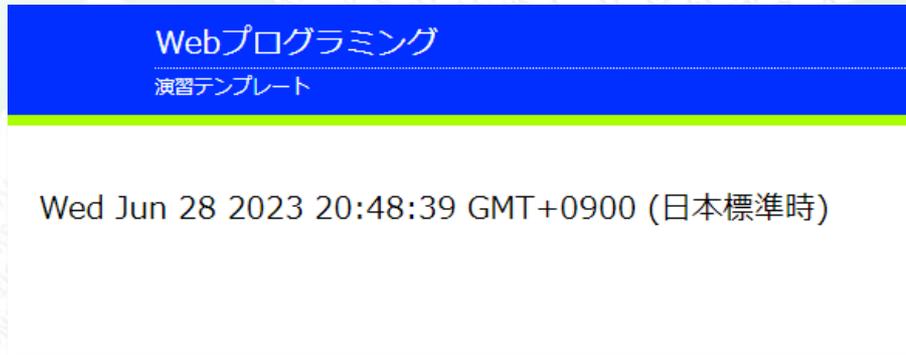
■ 四則演算以外の計算

- 円周率を表示する

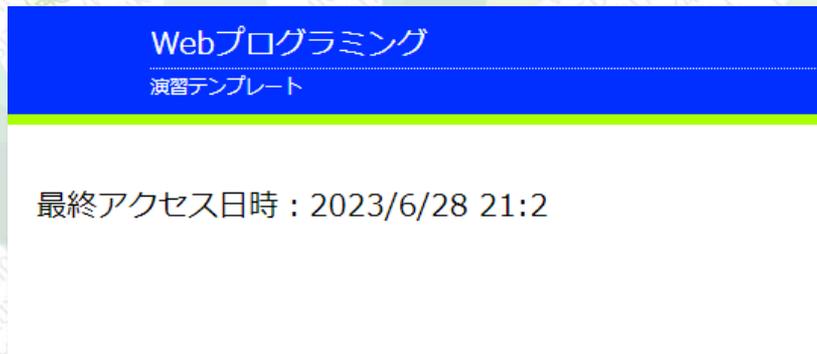
- Mathオブジェクト

日付と時間の処理

- 年月日と日時をわかりやすく表示する
 - 第8回で行った日付表示は少々わかりにくい



- この↑表記を「年/月/日 時:分」の形式で表示する
- Dateオブジェクトを用いて、個別に年、月、日、時、分を取得して処理する



日付と時間の処理

- 年月日と日時をわかりやすく表示する
 - テンプレートの「`index.html`」をコピーして、ファイル名を「`index1301.html`」に変更、下記を追記する
 - テンプレートの「`style.css`」もコピーする

```
18 <section>
19   <p>最終アクセス日時 : <span id="time"></span></p>
20 </section>
```

⋮

`index1301.html`

```
27 </footer>
28 <script>
29   'use strict';
30   const now = new Date();
31   const year = now.getFullYear();
32   const month = now.getMonth();
33   const date = now.getDate();
34   const hour = now.getHours();
35   const min = now.getMinutes();
36   const output = `${year}/${month + 1}/${date} ${hour}:${min}`;
37   document.getElementById('time').textContent = output;
38 </script>
```

日付と時間の処理

■ 年月日と日時をわかりやすく表示する

- ブラウザで確認すると、年月日に加えて24時間表記で日時が表示される

最終アクセス日時 : 2023/6/28 21:2

■ Dateオブジェクト

- Dateオブジェクトは**日時を扱うオブジェクト**であり、以下のような処理ができる
 1. 現在日時を取得する
 2. 過去や未来の(任意の)日時を設定する
 3. 日時の計算をする

日付と時間の処理

Dateオブジェクト

- Dateオブジェクトは初期化する(インスタンスを作成する)必要がある

```
const now = new Date();
```

← new でDateのインスタンスを作成、
変数 now に代入

Date(); の () が空の場合、**現在日時の情報**を記憶する

- 年、月、日などを個別に取得する

```
const year = now.getFullYear();
```

← 年を取得

```
const month = now.getMonth();
```

← 月を取得

```
const date = now.getDate();
```

⋮

```
const hour = now.getHours();
```

⋮

```
const min = now.getMinutes();
```

← 分を取得

各メソッドを呼び出して、値を取得し変数に代入する

.getMonth()は**1月~12月**が**0~11**の値で返されるので注意

日付と時間の処理

■ Dateオブジェクト

■ Dateオブジェクトの主なメソッド（値を取得する）

メソッド	説明
getFullYear()	年を取得する
getMonth()	月を0~11の数値で取得する(0が1月)
getDate()	日を取得する
getDay()	曜日を0~6の数値で取得する(0が日曜日)
getHours()	時を取得する
getMinutes()	分を取得する
getSeconds()	秒を取得する
getMilliseconds()	ミリ秒を0~999の数値で取得する
getTimezoneOffset()	時差を取得する
getTime()	1970/1/1の0時からの時間をミリ秒で取得する

日付と時間の処理

■ Dateオブジェクト

■ Dateオブジェクトの主なメソッド（値を設定する）

メソッド	説明
setFullYear(年)	年を設定する
setMonth(月)	月を0~11の数値で設定する(0が1月)
setDate(日)	日を設定する
setHours(時)	時を設定する
setMinutes(分)	分を設定する
setSeconds(秒)	秒を設定する
setMilliseconds(ミリ秒)	ミリ秒を0~999の数値で設定する
setTime(ミリ秒)	1970/1/1の0時からの時間をミリ秒で設定する

日付と時間の処理

Dateオブジェクト

■ 各変数の値を出力する

monthは(実際の月 - 1)の値が入っている
ので、1プラスする

バッククォート

```
const output = `${year}/${month + 1}/${date} ${hour}:${min}`;  
document.getElementById('time').textContent = output;
```

```
<section>
```

```
  <p>最終アクセス日時 : <span id="time"></span></p>
```

```
</section>
```

HTMLのid('time')を指定

HTMLの<p>タグで表示する

最終アクセス日時 : 2023/6/28 21:2

日付と時間の処理

Dateオブジェクト

12時間時計の表示にする

2023/7/28 13:57



2023/7/28 1:57p.m.

「index1301.html」に下記を追記・修正する

```
35 const min = now.getMinutes();
36 let ampm = '';
37 if(hour < 12) {
38     ampm = 'a.m.';
39 } else {
40     ampm = 'p.m.';
41 }
42 const output = `${year}/${month + 1}/${date} ${hour % 12}:${min}${ampm}`;
43 document.getElementById('time').textContent = output;
44 </script>
```

※ブラウザで表示を確認しましょう

`hour % 12`

hourには 0~23が入っている
12で割った余りにすると、12~23 が 0~11 になる

日付と時間の処理

Dateオブジェクト

- 任意の日時で初期化する
- 「index1301.html」に下記を追記する

index1301.html

```
29 'use strict';  
30 const now = new Date(2023, (12-1), 24, 17, 32);  
31 const year = now.getFullYear();
```

最終アクセス日時 : 2023/12/24 5:32p.m.

2023年 24日 17時 32分

```
const now = new Date(2023, (12-1), 24, 17, 32);
```

12月

月は、(月-1)の値を設定することに注意する

日付と時間の処理

■ Dateオブジェクト

■ 日数計算してみる

■ 「index1301.html」に下記を追記・修正する

```
29     'use strict';  
30     const day = new Date(2022, (12-1), 24, 17, 32);  
31     const now = new Date(2023, (12-1), 24, 17, 32);
```

⋮

```
41     ampm = 'p.m.';  
42     }  
43     //const output = `${year}/${month + 1}/${date} ${hour} : ${minute} : ${second}`;  
44     const output = `${(now - day)/86400000} 日`;  
45     document.getElementById('time').textContent = output;  
46 </script>
```

コメントアウト

index1301.html

追記

上記は、2022/12/24 から 2023/12/24 までの経過日数を計算している

Date型 - Date型 の計算結果はミリ秒単位で返されるので、それを1日分のミリ秒(24時間×60分×60秒×1,000ミリ秒=86,400,000)で割ると日数を取得できる

日付と時間の処理

■ Dateオブジェクト

■ 日数計算してみる

2022/12/24 から 2023/12/24 までの経過日数を計算した結果

```
const day = new Date(2022, (12-1), 24, 17, 32);  
const now = new Date(2023, (12-1), 24, 17, 32);  
  
(now - day)/86400000
```

最終アクセス日時 : 365 日

■ 他の日時指定の書き方

- ◆ 年月日は / 区切り、時分秒は : 区切りで指定できる

```
date = new Date('2023/12/24');
```

```
date = new Date('2023/12/24 17:32');
```

```
date = new Date('2023/12/24 17:32:45');
```

```
const day = new Date('2022/12/24');  
const now = new Date('2023/12/24 17:32:45');
```

四則演算以外の計算

- 円周率を表示する（小数点以下を切り捨てる）
- テンプレートの「index.html」をコピーして、ファイル名を「index1302.html」に変更、下記を追記する

index1302.html

```
18 <section>
19   <p>円周率は <span id="pi"></span> です。</p>
20   <p>小数点以下をすべて切り捨てると <span id="floor"></span> です。</p>
21 </section>
```

```
28 </footer>
29 <script>
30   'use strict';
31   document.getElementById('pi').textContent = Math.PI;
32   document.getElementById('floor').textContent = Math.floor(Math.PI);
33 </script>
34 </body>
```

円周率は 3.141592653589793 です。
小数点以下をすべて切り捨てると 3 です。

四則演算以外の計算

■ 円周率を表示する（小数点以下を切り捨てる）

■ Mathオブジェクト

- ◆ 数学関係の定数やメソッドが利用できる

Math.PI ... 円周率

<p>円周率は

```
document.getElementById('pi').textContent = Math.PI;
```

HTMLのid('pi')を指定

Math.floor(数値) ... 小数点以下を切り捨てる

<p>小数点以下をすべて切り捨てると

```
document.getElementById('floor').textContent = Math.floor(Math.PI);
```

HTMLのid('floor')を指定

四則演算以外の計算

- 円周率を表示する（小数第0位まで表示）
- 小数点第2位まで表示できるようにする
- 「index1302.html」に下記を追記する

index1302.html

```
18 <section>
19   <p>円周率は <span id="pi"></span> です。</p>
20   <p>小数点以下をすべて切り捨てると <span id="floor"></span> です。</p>
21   <p>小数第3位以下を切り捨てると <span id="output"></span> です。</p>
22 </section>
```

:

```
32 document.getElementById('pi').textContent = Math.PI;
33 document.getElementById('floor').textContent = Math.floor(Math.PI);
34 function point(num, digit) {
35   const mover = 10 ** digit;
36   return Math.floor(num * mover) / mover;
37 }
38 document.getElementById('output').textContent = point(Math.PI, 2);
39 </script>
```

小数第3位以下を切り捨てると 3.14 です。

四則演算以外の計算

- 円周率を表示する（小数第0位まで表示）
 - 小数点第2位まで表示できるようにする
 - Mathオブジェクトでは、**小数点の桁数を指定して切り捨てるメソッドがない** → ちょっと工夫が必要
 - 小数点を含む値 X を小数点第 N位まで表示するには

＜ X=3.1415926 を 小数点第 N=2 位まで表示する例＞

※10のN乗は **10**N** と書く

1. $p = 10$ のN乗 $p = 10$ の2乗 = 100
2. $a = X * p$ を計算 $a = 3.1415926 * 100 = \underline{314.15926}$
3. $b = \text{Math.floor}(a)$.. $b = \text{Math.floor}(314.15926) = \underline{314}$
4. $c = b / p$ を計算 $c = 314 / 100 = 3.14$
5. c を表示 3.14

四則演算以外の計算

- 円周率を表示する（小数第0位まで表示）
 - 小数点第2位まで表示できるようにする
 - function point()を呼び出すと、指定した桁数で小数点以下を切り捨てる

```
extContent = point(Math.PI, 2);
```

それぞれの値が代入される

```
function point(num, digit) {  
  const mover = 10 ** digit;  
  return Math.floor(num * mover) / mover;  
}
```

1. $p = 10$ のN乗、moverは100が入る

2. $a = X * p$ を計算

3. $b = \text{Math.floor}(a)$ 、小数点切り捨て

4. $c = b / p$ を計算

四則演算以外の計算

■ Mathオブジェクトの主なプロパティ

プロパティ	説明
Math.PI	円周率、3.141592658979・・・
Math.SQRT1_2	1/2の平方根
Math.SQRT2	2の平方根、1.41421356

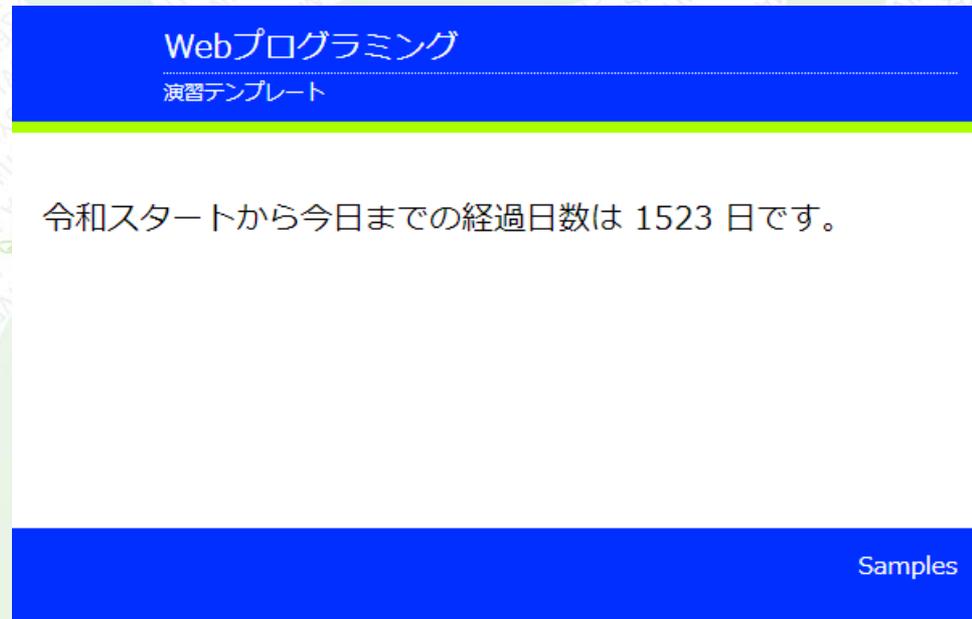
■ Mathオブジェクトの主なメソッド

メソッド	説明
Math.abs(x)	xの絶対値
Math.floor(x)	xの小数点以下を切り捨て、xより小さい整数にする
Math.trunc(x)	xの小数点以下を切り捨てる
Math.sin(x)	xのサイン
Math.cos(x)	xのコサイン
Math.tan(x)	xのタンジェント

メソッド	説明
Math.pow(x,y)	xのy乗
Math.sqrt(x)	xの平方根
Math.random()	0以上1未満の乱数
Math.max(a,b,...)	a,b...のうち最大値を返す
Math.min(a,b,...)	a,b...のうち最小値を返す
Math.round(x)	xの小数点以下を四捨五入する

演習 1

テンプレートの「[index.html](#)」をコピーして、ファイル名「[ensyu01.html](#)」に変更しなさい。
令和スタートの日(2019年5月1日)から、今日までの経過日数を表示するようにしなさい。



<実行例>

演習 2

テンプレートの「`index.html`」をコピーして、ファイル名「`ensyu02.html`」に変更しなさい。
数値を入力して、その数値を半径とする円の面積を計算して表示する。入力はプロンプト(第9回の講義資料を参照)で行うものとし、実行例のようにHTMLで結果を表示するようにしなさい。また、計算結果は小数点第3位以降を切り捨てて、小数点第2位までを表示すること。

このページの内容:
半径を入力してください

OK キャンセル



Webプログラミング
演習テンプレート

半径 1 の円の面積は 3.14 です。

Samples

このページの内容:
半径を入力してください

OK キャンセル



Webプログラミング
演習テンプレート

半径 12 の円の面積は 452.38 です。

Samples

<実行例>

演習 3

テンプレートの「[index.html](#)」をコピーして、ファイル名「[ensyu03.html](#)」に変更しなさい。

xとyの値を入力し、(0,0)-(x,y)の2点間の距離を計算して表示しなさい。(距離の計算式は $\sqrt{x^2 + y^2}$)

なお、値の入力はformのテキストフィールド2個とボタンを使用すること(第12回の講義資料を参照)。

Webプログラミング
演習テンプレート

(0,0)-(1, 1)の距離は 1.414 です。

Samples

Webプログラミング
演習テンプレート

(0,0)-(3, 4)の距離は 5 です。

Samples

<実行例>