

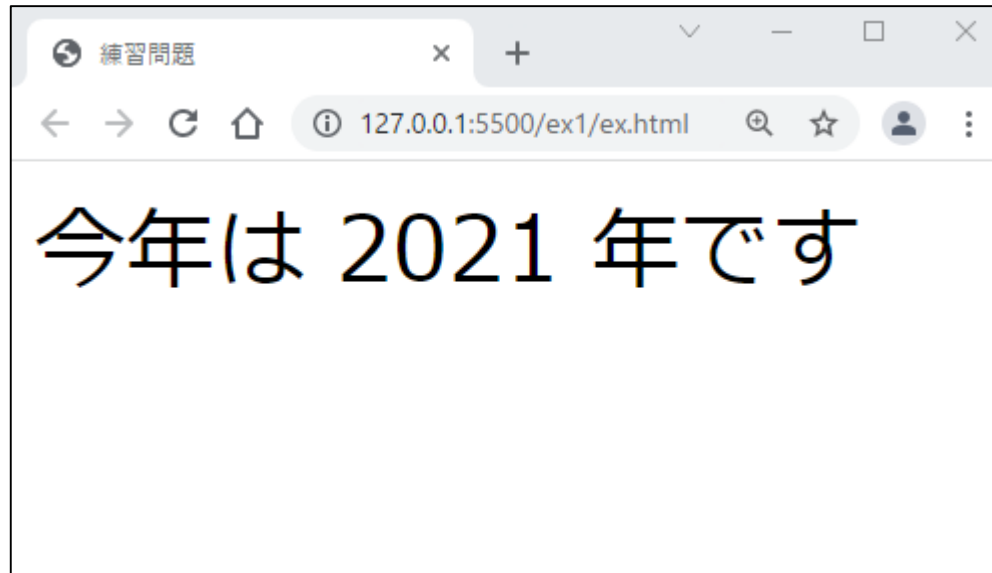
2-04 step1を終えたら...

---

# 練習問題(ex1-1)



HTMLの要素を使用せず「今年はXXXX年です」と表示。

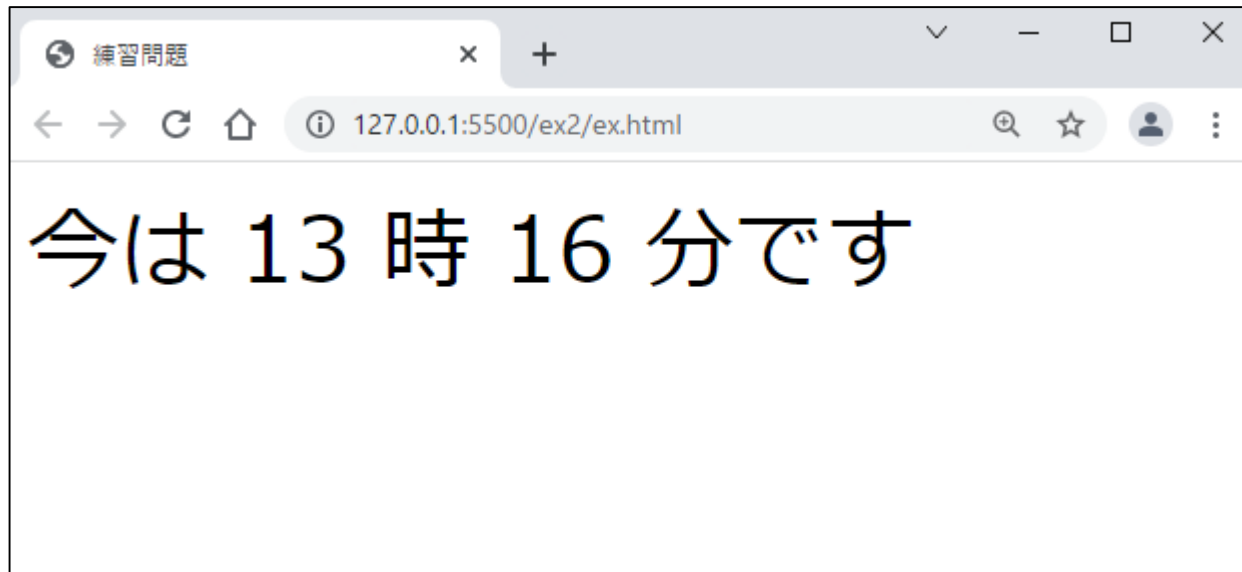


ヒント : Dateオブジェクト(p169)、writelnメソッド

# 練習問題(ex1-2)



HTMLの要素を使用せず、現在時刻を表示。

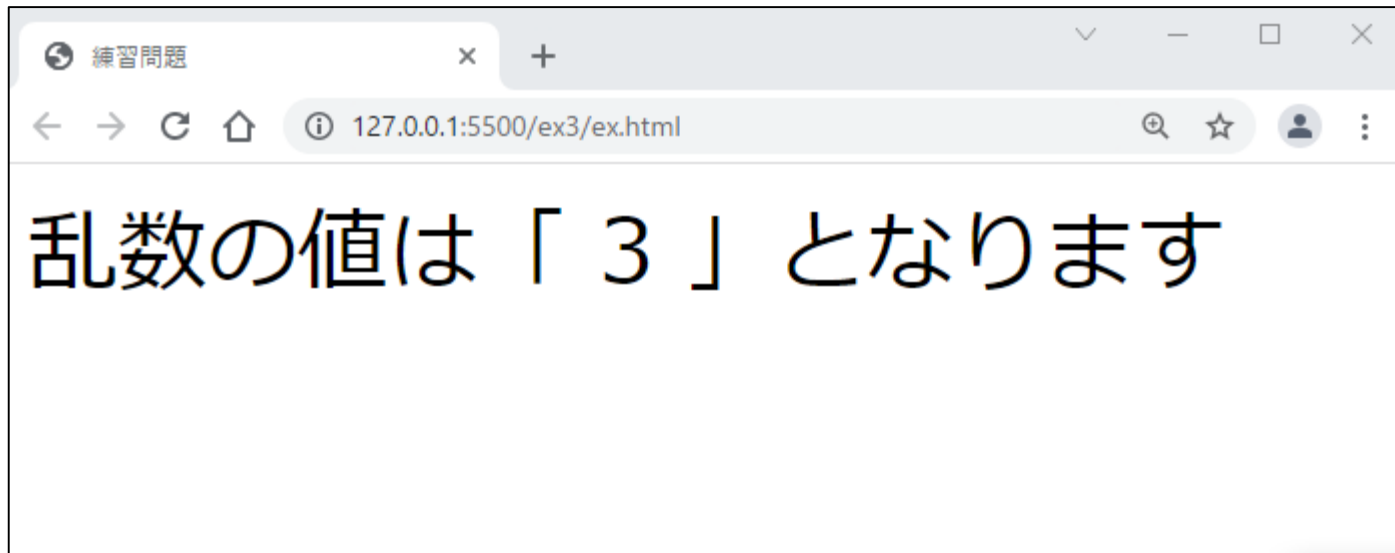


ヒント : Dateオブジェクト(p169)、writelnメソッド

# 練習問題(ex1-3)



Mathオブジェクトを使用し、1～6をランダムに表示。



ヒント : Mathオブジェクト(p180)

3-01 step2を終えたら...

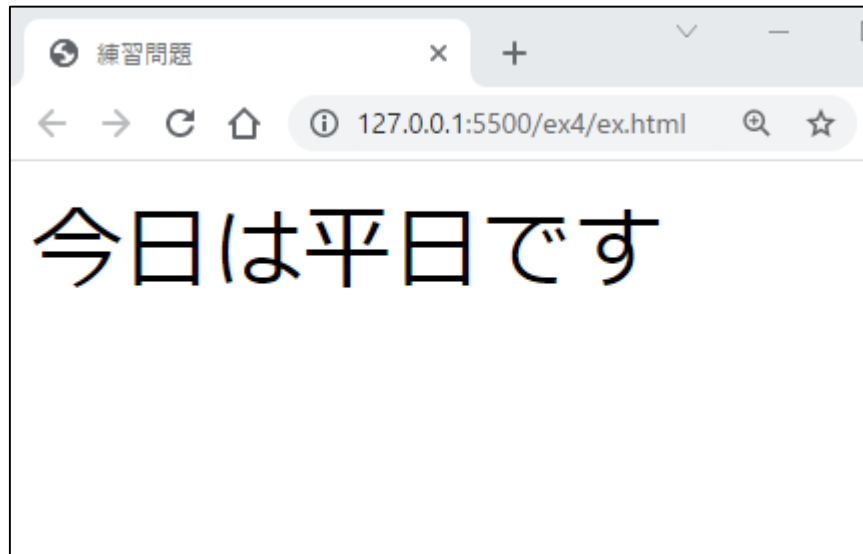
---

# 練習問題(ex2-1)



現在の日付を取得し、平日であれば「今日は平日です」、休日であれば「今日は休日です」と表示。

月曜日～金曜日の場合



土、日曜日の場合



ヒント : Dateオブジェクト(p169) 、 if文

# 練習問題(ex2-2)

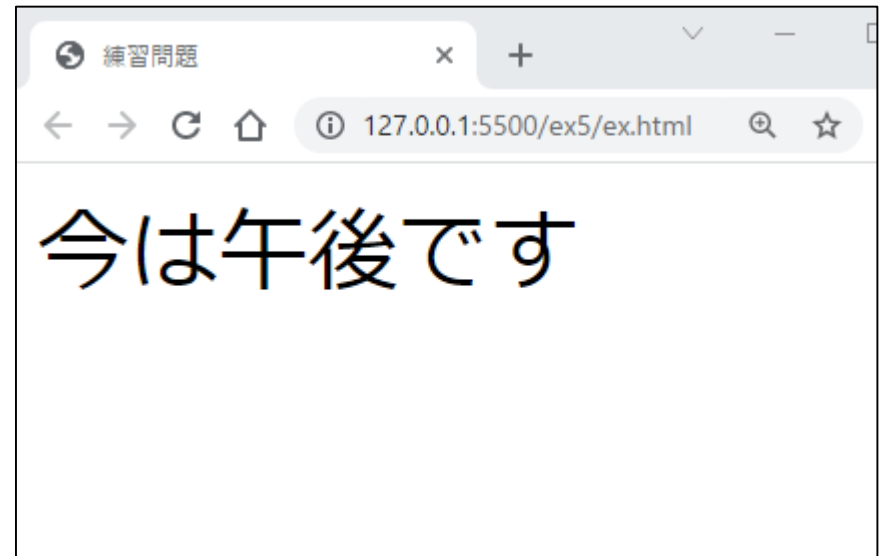


現在の時刻を取得し、午前であれば「今は午前です」、午後であれば「今は午後です」と表示。

0時～11時の場合



12時～23時の場合



ヒント : Dateオブジェクト(p169) 、 if文

3-04 step1を終えたら...

---



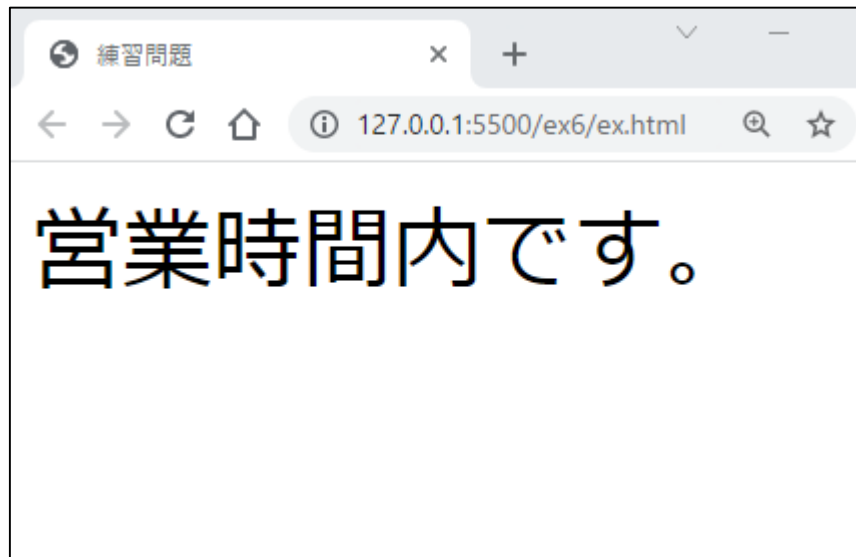
# 練習問題(ex3-1)



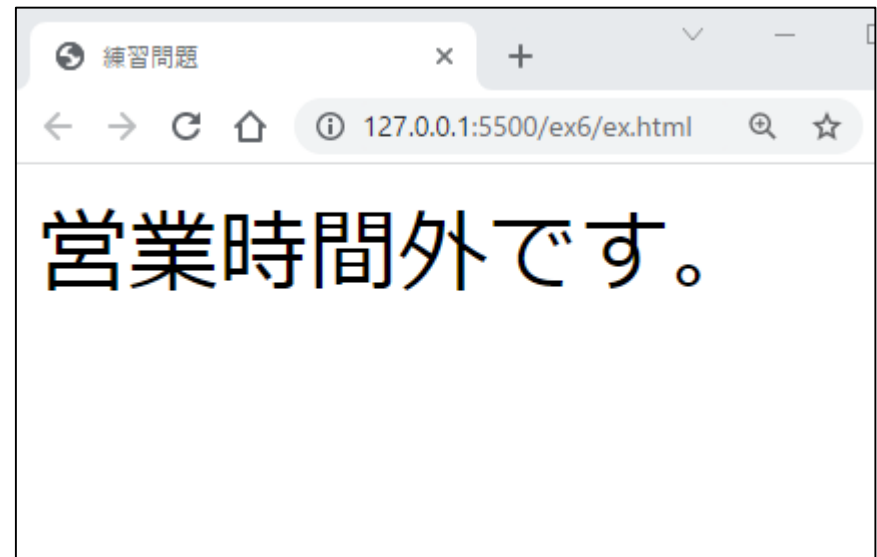
現在日時を取得し、

営業時間(9:00~16:59)に合わせて表示する内容を変更。

営業時間内(9:00~16:59)の場合



営業時間外(17:00~8:59)の場合



ヒント : Dateオブジェクト(p169) 、 if文、 定数、 比較演算子

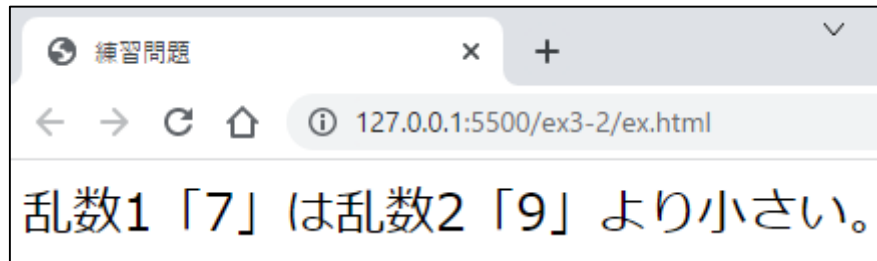
# 練習問題(ex3-2)



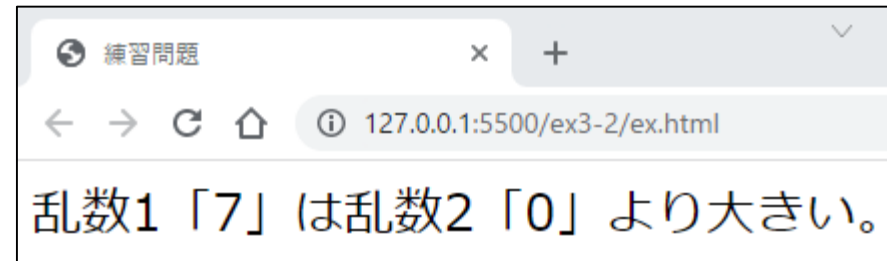
整数の乱数を2つ用意し、値を比較して結果を表示。

乱数の範囲：0～9

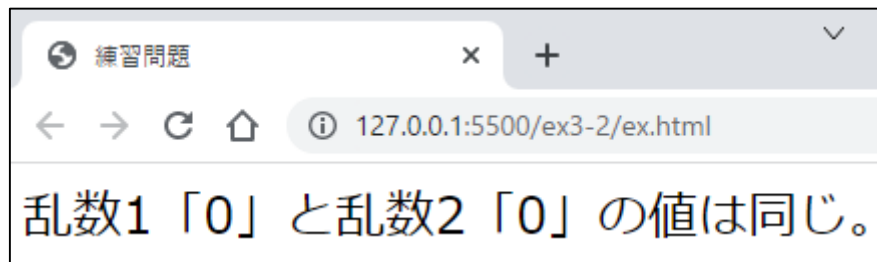
どちらかが小さい場合



どちらかが大きい場合



同じ値の場合

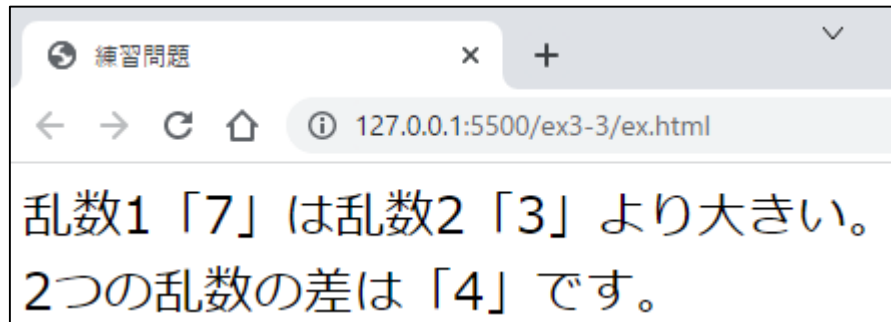


ヒント：Mathオブジェクト(p180)、if文

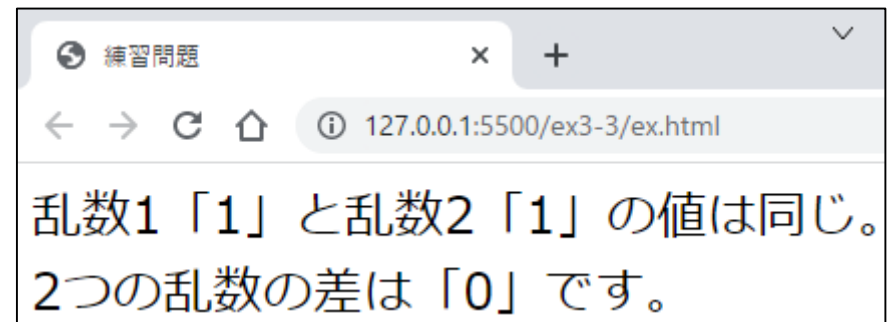
ex3-2の内容に加え、

その時どのくらいの差があるのかも表示

どちらかが小さい・大きい場合



同じ値の場合



ヒント : Mathオブジェクト(p180) 、 Math.absメソッド

ダイアログで3つ数値を入力し、最大値・最小値を表示。  
その時、入力された値が数値でない場合  
「数字を入力してください。」と表示。

数字を入力してね。

OK キャンセル

3つすべて数字を入力した場合

練習問題 x +

← → ↺ 🏠 ⓘ 127.0.0.1:5500/ex3-4/ex.html

最大値は32です。  
最小値は13です。

数字以外が入力された場合

練習問題 x +

← → ↺ 🏠 ⓘ 127.0.0.1:5500/ex3-4/ex.html

1つ目は数字ではありませんでした。  
数字を入力してください。  
3つ目は数字ではありませんでした。  
数字を入力してください。  
最大値はNaNです。  
最小値はNaNです。

ヒント :promptメソッド、Number.isNaNメソッド、Mathオブジェクト(p180)

引数が NaN (Not a Number : 数値でない) かどうかを判定。

数値でないとき、trueを返す。

(類似のものに「typeof 演算子」がある。)

データの型を調べる演算子。

「**typeof 値 または 変数名**」で  
型の名前が文字列で返される。

index.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>typeof演算子</title>
</head>
<body>
  <p>コンソールで動作確認できます。</p>
  <!-- jsファイルを読み込み -->
  <script src="index.js"></script>
</body>
</html>
```

```
'use strict'

// 数字
console.log('10 => ' + typeof 10);

// 文字列
console.log("'aaa' => " + typeof 'aaa');

// String型へ変換後の値
console.log('String(10) => ' + typeof String(10));

// 定義済みの定数・変数
let def = 123;
console.log('let def = 123 => ' + typeof def);

// 未定義の定数・変数
let undef;
console.log('let undef => ' + typeof undef);

// 配列
console.log('[1,2,3] => ' + typeof [1,2,3]);
```

```
// オブジェクト
console.log("{num: 1, str: 'aaa'} => " + typeof {num: 1, str: 'str'});

// Stringオブジェクト
console.log('new String() => ' + typeof new String());

// Arrayオブジェクト
console.log('new Array() => ' + typeof new Array());

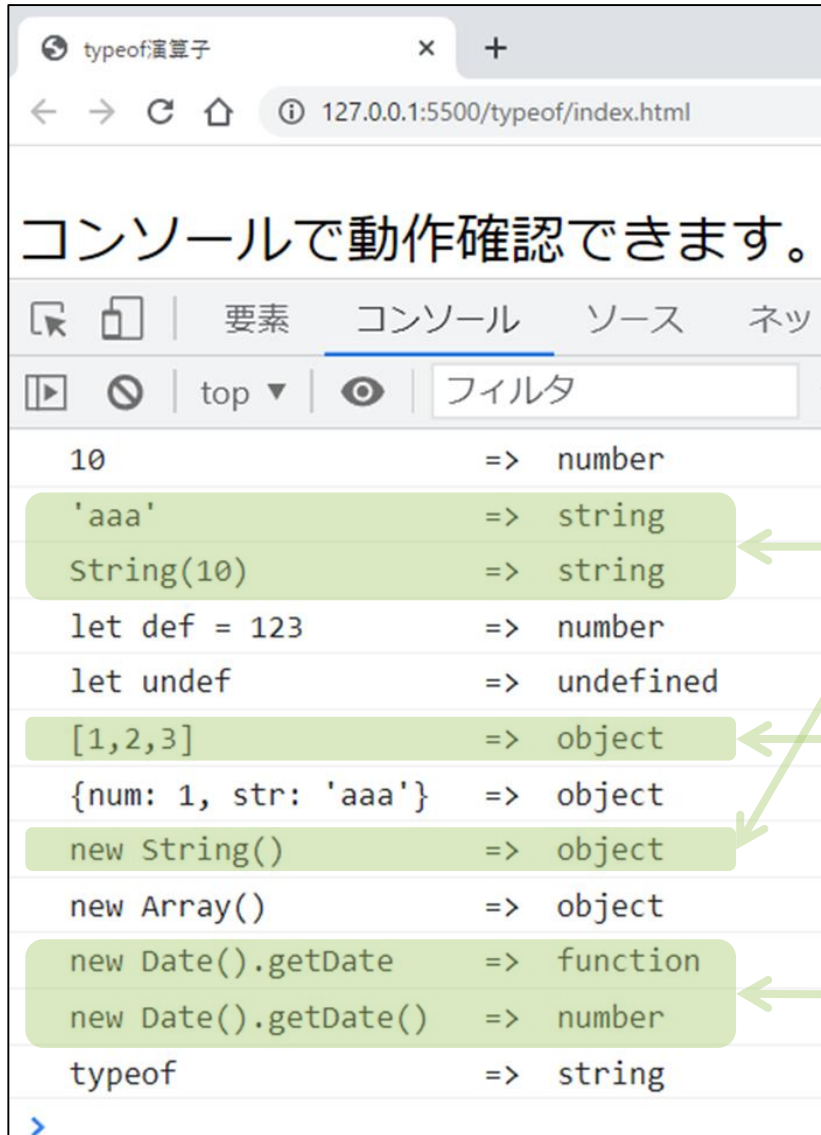
// 関数(ファンクション)
console.log('new Date().getDate => ' + typeof new Date().getDate);

// ファンクションから受け取った値
console.log('new Date().getDate() => ' + typeof new Date().getDate());

// typeof
console.log('typeof => ' + typeof typeof 10);
```



## 実行結果



The screenshot shows a web browser window with the title "typeof演算子" and the address "127.0.0.1:5500/typeof/index.html". The console is open, showing the following results:

Expression	Result
10	number
'aaa'	string
String(10)	string
let def = 123	number
let undef	undefined
[1,2,3]	object
{num: 1, str: 'aaa'}	object
new String()	object
new Array()	object
new Date().getDate	function
new Date().getDate()	number
typeof	string

string(データ型) と

String(オブジェクト型)は結果が異なる

配列のデータ型は「object」として返される

「関数名」 は「関数(function)」、

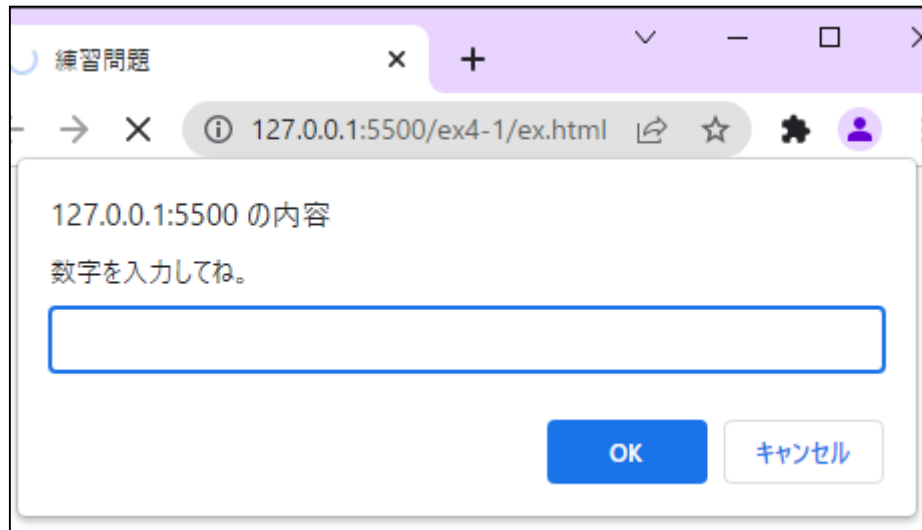
「関数名()」は「関数の戻り値」の型を返している。

3-08 step2を終えたら...

---

3つの整数を入力してもらい、値を比較して結果を  
コンソールへ表示。(数値以外の場合は対策しなくても可)

数値を入力



A screenshot of a web browser window. The address bar shows '127.0.0.1:5500/ex4-1/ex.html'. A modal dialog box is open with the title '127.0.0.1:5500 の内容' and the text '数字を入力してね。' (Please enter a number). There is a text input field and two buttons at the bottom: 'OK' and 'キャンセル' (Cancel).

2つの値が等しい場合



A screenshot of a web browser window showing the result of a comparison. The address bar shows '127.0.0.1:5500/ex4-1/ex.html'. The page content displays the following text:

Default levels ▼	No Issues
1つ目 : 23	
2つ目 : 89	
3つ目 : 23	
2つの値は等しいです	

ヒント : promptメソッド、if文

# 練習問題(ex4-2)



整数1つ入力し、数に応じて

+ - を繰り返しコンソールへ出力

表示する個数の入力

127.0.0.1:5500 の内容

数字を入力してね。

OK キャンセル

2の場合

Default levels ▼ No Issues

入力された値 : 2

+

-

5の場合

Default levels ▼ No Issues

入力された値 : 5

+

-

+

-

+

ヒント : promptメソッド、if文、繰り返し文

整数2つ入力し、2つの値のうち小さい方を返す  
ファンクションを作成。入力した値とファンクションの  
戻り値はwebページ上に出力。(Math.min())は利用しない)

## 値の入力

127.0.0.1:5500 の内容

数字を入力してね。

## 出力例

練習問題

127.0.0.1:5500/ex4-3/ex.html

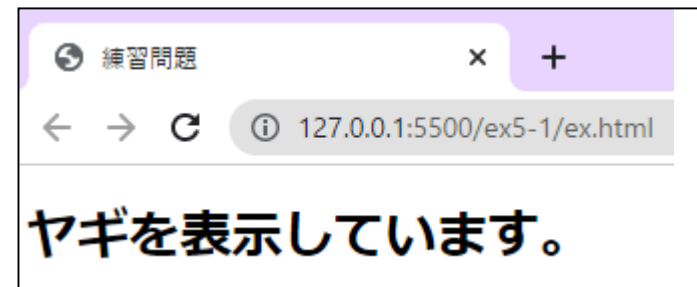
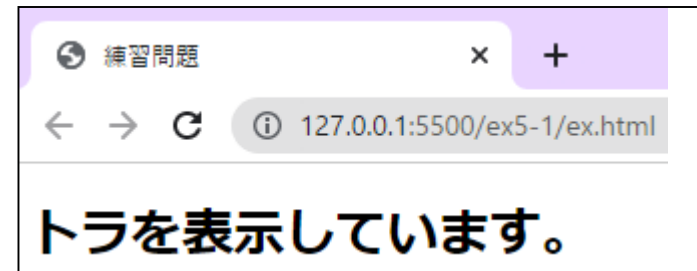
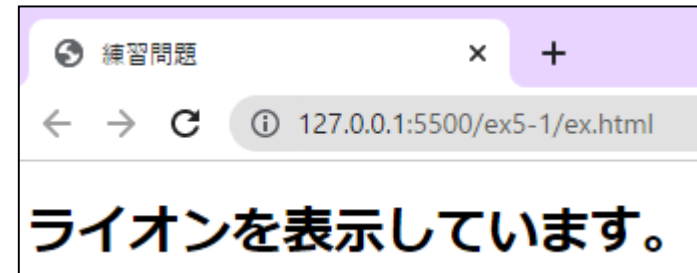
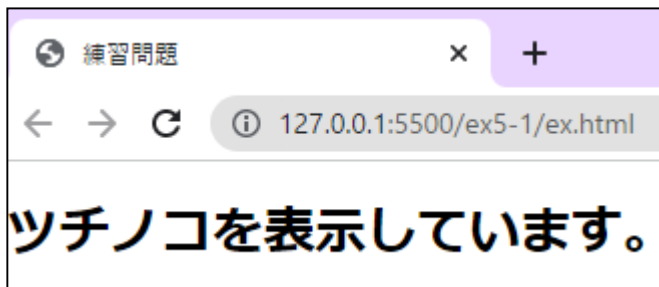
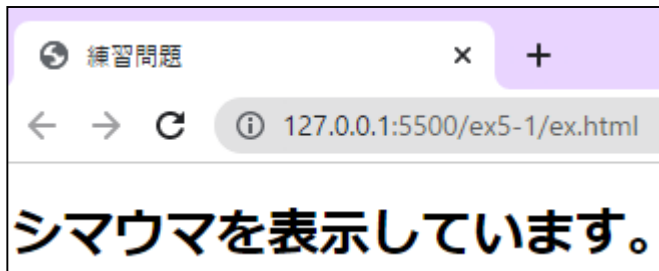
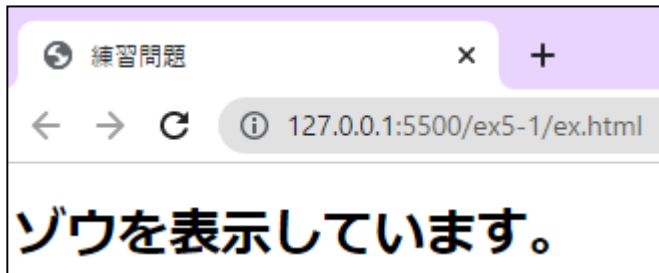
入力された値1 : 5  
入力された値2 : 3  
小さい方の値は3です。

ヒント : promptメソッド、ファンクションの扱い方、if文

3-10 step5を終えたら...

---

ランダムで動物の名前を表示(参考 : h1タグで実装)  
配列を作成し、動物の名前を追加。  
webページにランダムで動物の名前を表示。



ヒント : 配列の扱い方、乱数、要素の取得、テンプレートリテラル(文字列)

3-11 step3を終えたら...

---



# 練習問題(ex6-1)

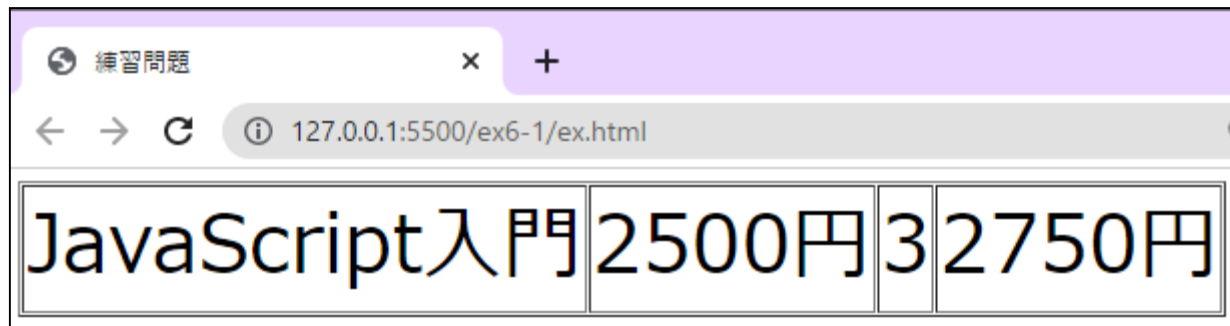


jsbookオブジェクトにaddtaxプロパティを追加しtableで表示。  
addtaxはパラメータに消費税率(10%)をかける  
ファンクションを持つ。

addtaxプロパティを追加する前のjsbookオブジェクト

```
// オブジェクトの作成
let jsbook = { title: 'JavaScript入門',
               price: 2500,
               stock: 3
             };
```

出力例



ヒント: プロパティの扱い方、要素の取得

4-01 step2を終えたら...

---

HTMLでテキストボックス、ラジオボタン、送信ボタンを実装し、送信ボタンを押下するとアラートで入力、選択した内容を表示

HTMLの例

A screenshot of a web browser window showing a form. The browser's address bar displays '127.0.0.1:5500/ex7-1/ex.html'. The form contains a text input field labeled '名前' (Name), three radio buttons labeled '炎属性' (Fire Attribute), '水属性' (Water Attribute), and '草属性' (Grass Attribute), and a '送信' (Send) button. The '炎属性' radio button is selected.

出力例

A screenshot of the same web browser window, but the text input field now contains 'トカゲ' (Tokage) and the '炎属性' radio button remains selected. The '送信' button is still present.



127.0.0.1:5500 の内容  
入力された名前は「トカゲ」、  
選択された属性は「炎属性」。

OK

ヒント :基本動作のキャンセル、要素の取得、window.alertメソッド

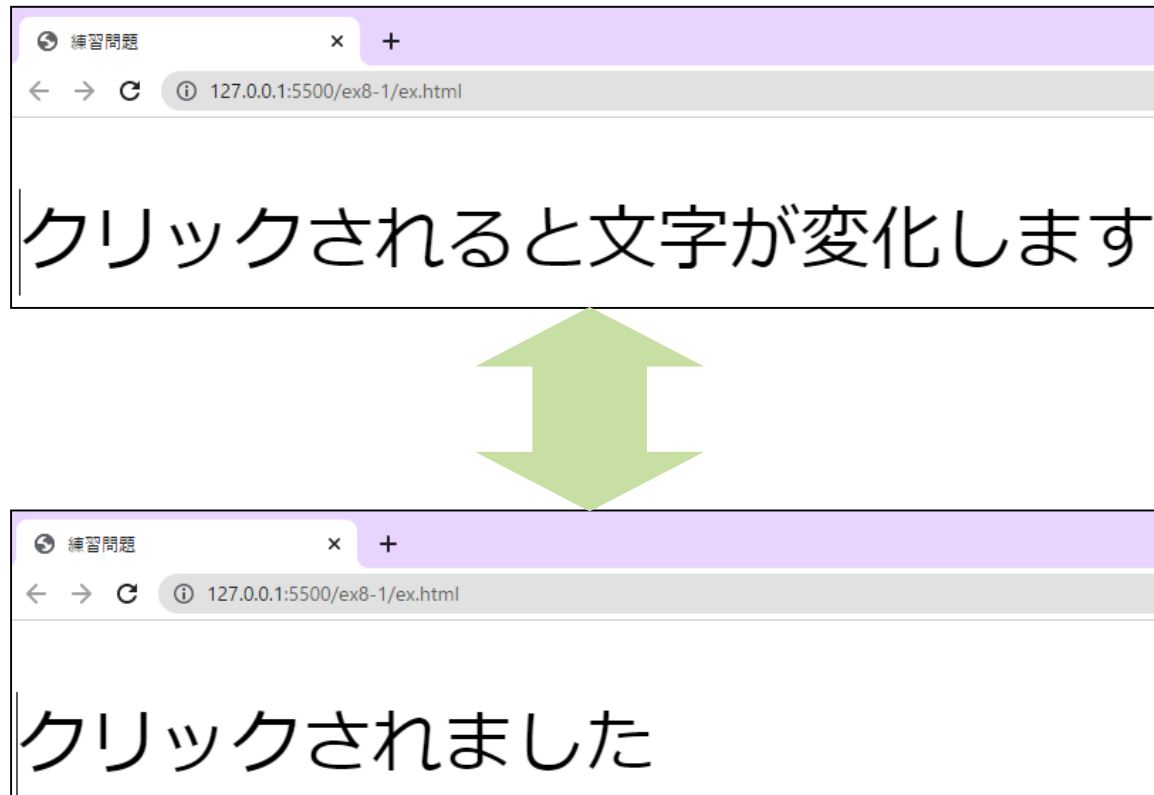
6-02 step3を終えたら...

---

テキストをクリックするとテキスト内容が変更される。

さらに、もう一度テキストをクリックすると内容が元に戻る。

出力例



ヒント : jQuery、切り替え状態の管理

6-03 step2を終えたら...

---

jsonファイルからデータを読みだして  
テーブル上に内容を表示

記述するプロパティ名と値

プロパティ名	値
title	JS入門
price	1000
stocks	100

出力例



タイトル	税抜価格	在庫数
JS入門	1000円	100冊

ヒント : jsonファイルの記述、\$.ajaxメソッド、.doneメソッド、.failメソッド