




# SIJP コーディング キャンプ



## JavaScript 編



プログラミングの準備をしよう！





---

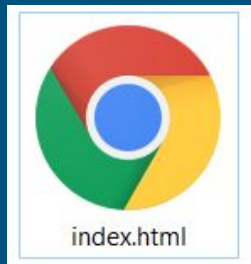
# USBドライブをコンピュータにつなげよう！



# チュートリアルをブラウザに表示しよう！

 File Explorer

 USB Drive (E:)  
 codingcamp  
 greetingcard  
 tutorial



## メリークリスマス！



⇐アイコンをクリックしてね

作りたかったグリーティングカードは  
こんな感じですが・・・

メリークリスマス！



⇐アイコンをクリックしてね

なんだか、いろいろ変・・・

メリークリスマス！



⇐アイコンをクリックしてね

どこにバグがあるか、わかるかな？

# バグを直そう！

---

バグを直すことをデバッグと言います。

ソースコードを見たい！  
ブラウザの開発者ツールを使う

資料1:「資料1 Chrome ブラウザ 開発ツール」

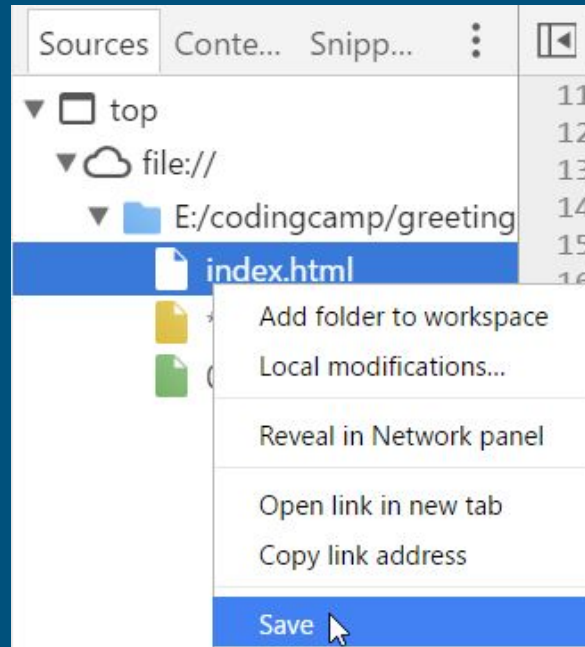


言葉の間違いを直してみる...

1. 「メリークリスマス！」に直す
2. 「Save」メニューで保存しようとするけど...

1

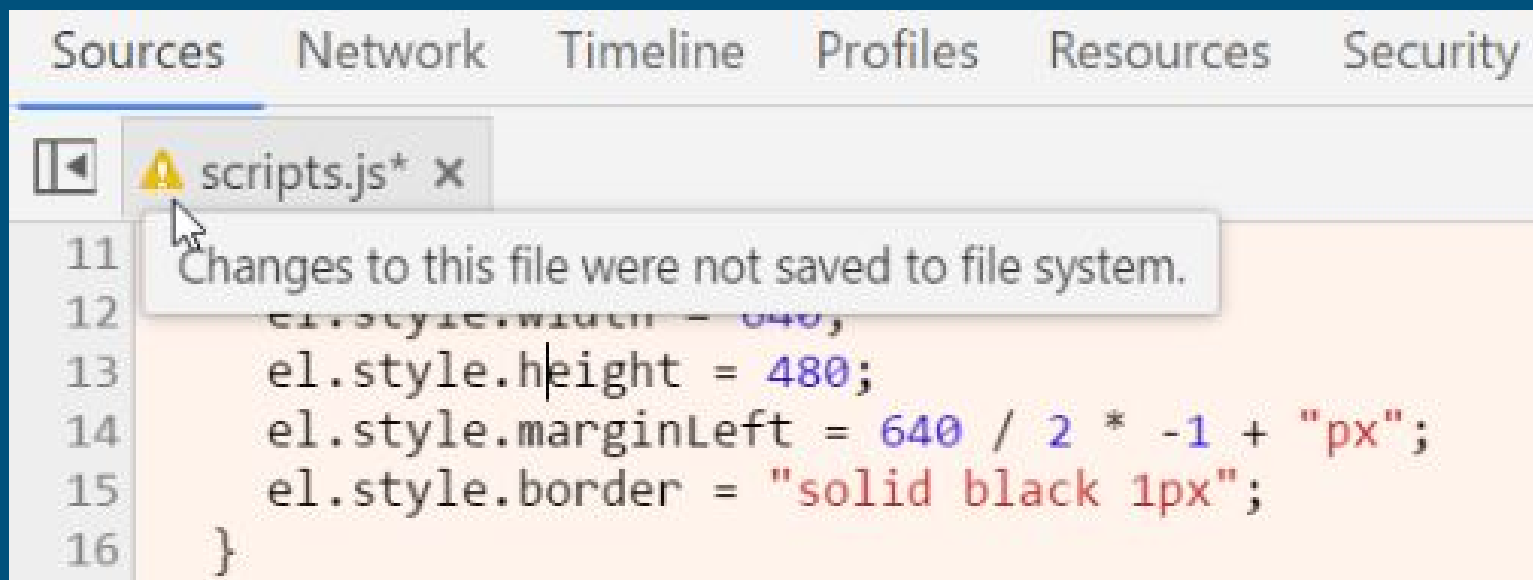
```
el.textContent = "メリークリスマス！";
```



2

「保存出来ません」とエラーが出ます。

“Changes to this file were not saved to *file system*.”

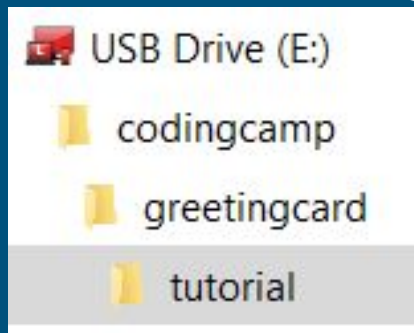


File system ってなに？

# Webサーバー (file system)



READ-ONLYの  
コピーを送信



変更しても

保存できない！

でも、安心してください



ブラウザからソースコード  
を変更できるようにする！

資料2:「Chrome ブラウ  
ザーからソースコードを変更  
できるようにする」

プログラミングを始めよう！



バグ1。グリーティングの文字と場所がおかしい！



メリークリスマス



←アイコンをクリックしてね

# 文字のまちがいをなおそう！

```
23     el.style.fontSize = "60px";  
24     el.textContent = "メリークリスマス！";  
25 }
```



```
23     el.style.fontSize = "60px";  
24     el.textContent = "メリークリスマス！";  
25 }
```

直したら、保存して、ページを再度読み込み  
しよう。正しく修正されているか確認。

step01

# left と top の数を変えよう！

```
20 el.style.position = "absolute";  
21 el.style.left = 0;  
22 el.style.top = 0;  
23 el.style.fontSize = "60px";  
24 el.textContent = "メリークリスマス！";
```



```
20 el.style.position = "absolute";  
21 el.style.left = 100;  
22 el.style.top = 120;  
23 el.style.fontSize = "60px";  
24 el.textContent = "メリークリスマス！";
```

ところで left と top って何？

step01



0



5



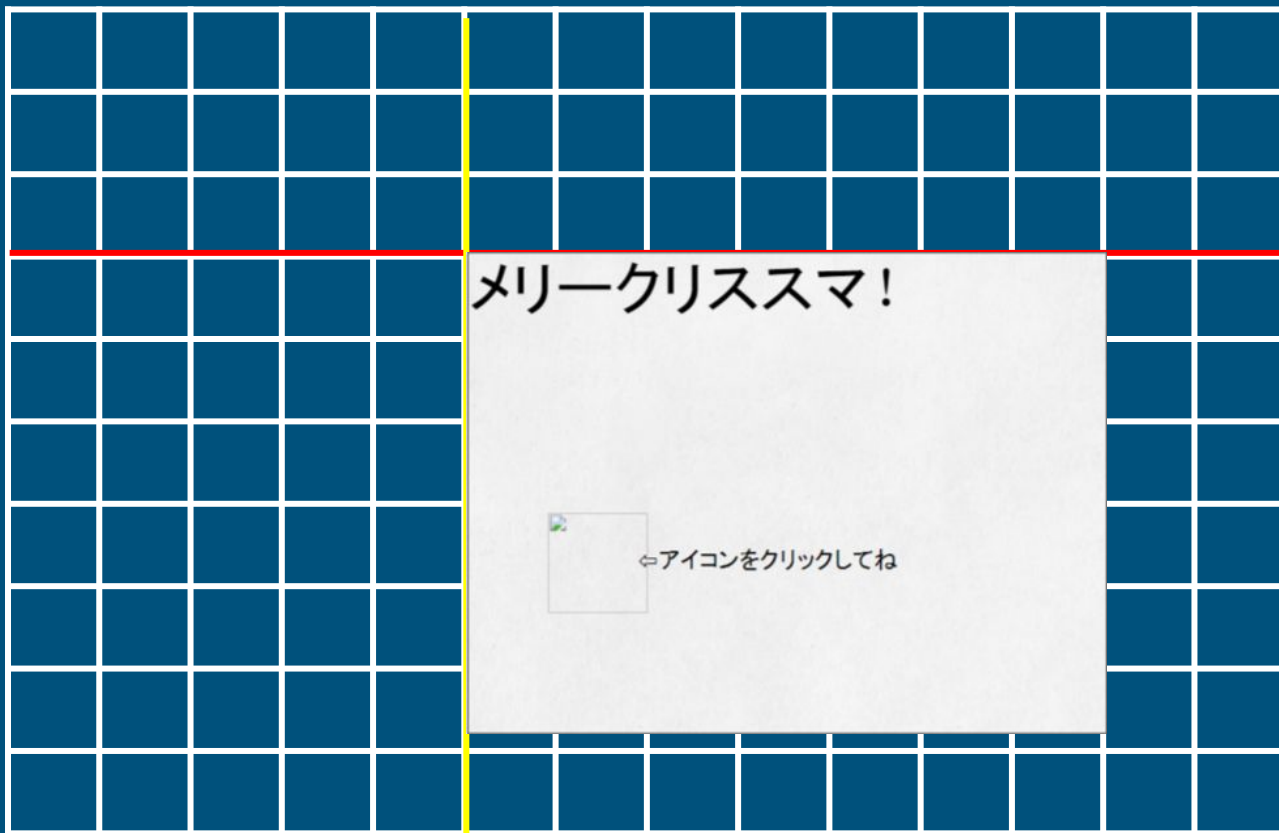
10

top=3

0



5



メリークリスマス!

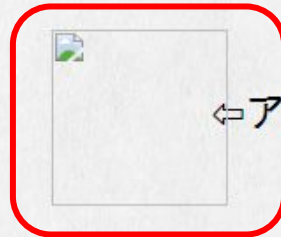


アイコンをクリックしてね

left=5

## バグ2。サンタさん、どこ行った？

メリークリスマス！



⇐アイコンをクリックしてね

# 正しいファイル名にしよう！

```
42 document.getElementById("card").appendChild(el);  
43 el.src = "santa.png";  
44 el.style.position = "absolute";
```



santa\_claus.png

```
42 document.getElementById("card").appendChild(el);  
43 el.src = "santa_claus.png";  
44 el.style.position = "absolute";
```

# バグ3。カードにカゲがない！



メリークリスマス！



←アイコンをクリックしてね

# カゲのスタイルをつかおう！

```
14 el.style.marginLeft = 640 / 2 * -1 + "px";  
15 el.style.border = "solid black 1px";  
16 }
```



`el.style.boxShadow = "10px 10px 5px #888888";`

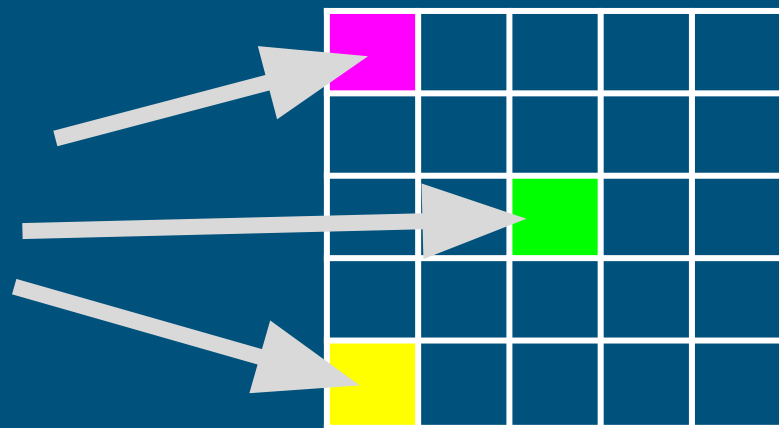
```
14 el.style.marginLeft = 640 / 2 * -1 + "px";  
15 el.style.boxShadow = "10px 10px 5px #888888";  
16 }
```

step03

ところで px って何？



ところで・・・5px の「px」てなに？

”ピクセル”








5px = 粒5つ分

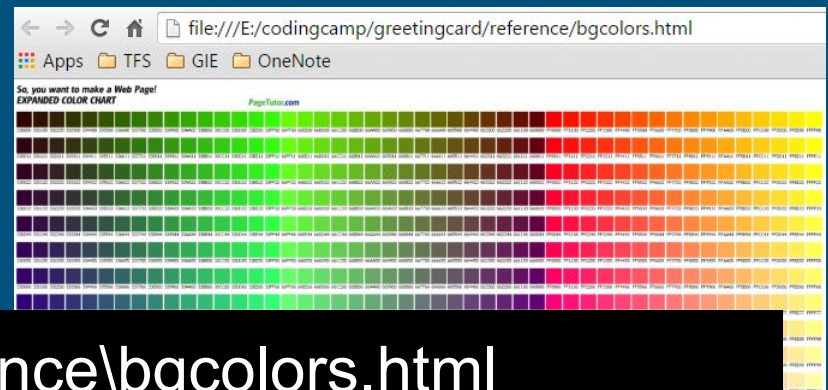
# #888888 ← なんだこりゃ？

	#000000
	#808080
	#FFFFFF

コンピューター上  
で色を表す

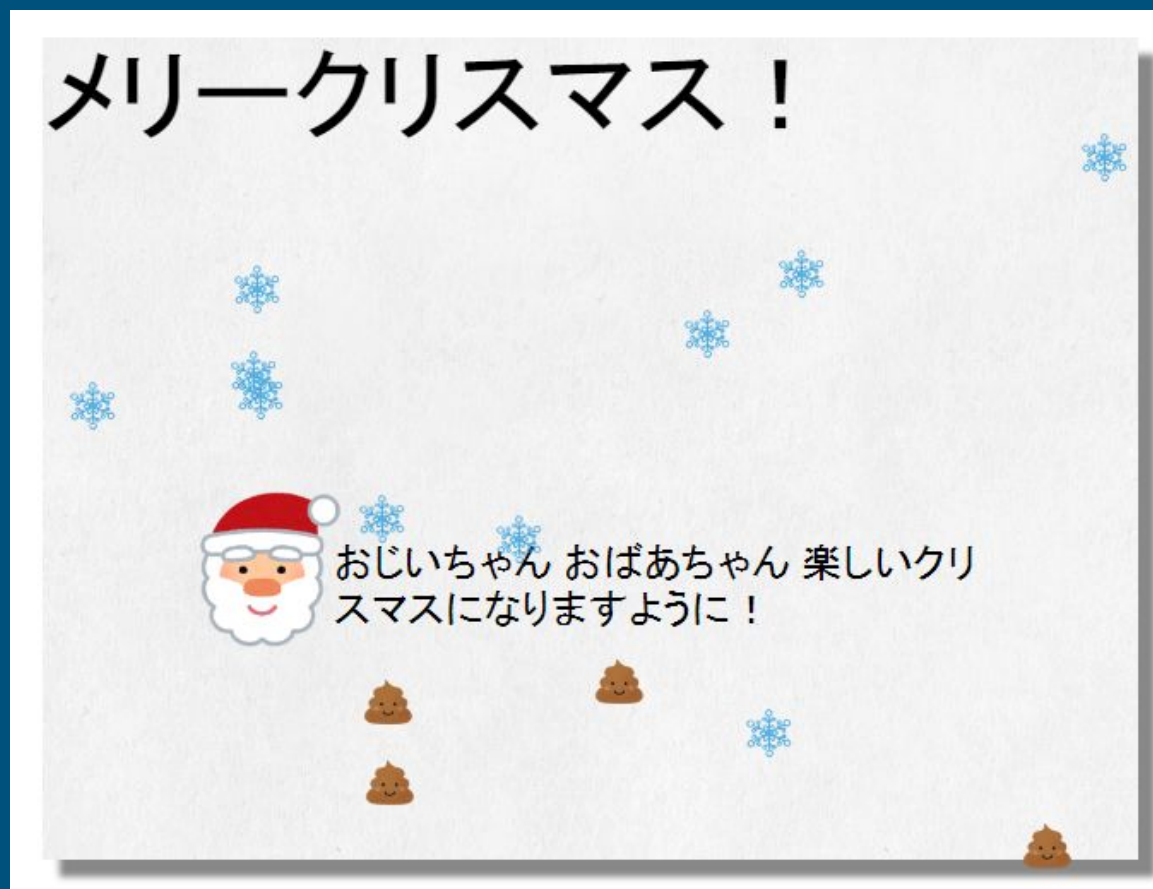
	#FF0000
	#FFFF00
	#00FF00
	#00FFFF
	#0000FF

他に、どんな色が使え  
るか見てみよう



codingcamp\greetingcard\reference\bgcolors.html

バグ4。サンタを10回以上クリックすると・・・変な画像が・・・





とりあえず、ウンチはいらない(笑)

```
66     if (clickCount < 10) {  
67         im.src = "snow_crystal2.png";  
68     } else {  
69         im.src = "unchi_character.png"  
70     }
```



ファイルの名前を変える

```
66     if (clickCount < 10) {  
67         im.src = "snow_crystal2.png";  
68     } else {  
69         im.src = "snow_crystal3.png"  
70     }
```

step04

10回ではなく5回で画像が変わるようにしよう。

```
66     if (clickCount < 10) {  
67         im.src = "snow_crystal2.png";  
68     } else {  
69         im.src = "snow_crystal3.png"  
70     }
```



数字を10から5に変える

```
66     if (clickCount < 5) {  
67         im.src = "snow_crystal2.png";  
68     } else {  
69         im.src = "snow_crystal3.png"  
70     }
```

step04

# if文をいろいろ試してみよう

```
(clickCount > 5)
```

```
(clickCount == 3)
```

```
(clickCount >= 5)
```

```
(clickCount % 2 == 0)
```

# バグ5。「おばさん」が表示されない。

おじいちゃん  
おばあちゃん  
おばさん

3人にカードをおくるつもりなのに、「おばさん」の文字  
がでない。



おじいちゃん おばあちゃん 楽しいクリ  
スマスになりますように！

ソースコードに「おばさん」はあるのに...

```
3 var clickCount = 0;  
4 var persons = ['おじいちゃん', 'おばあちゃん', 'おばさん'];
```

# "おばさん"のデータを加えよう

```
49 var to = "";  
50 to += persons[0] + " ";  
51 to += persons[1] + " ";  
52
```

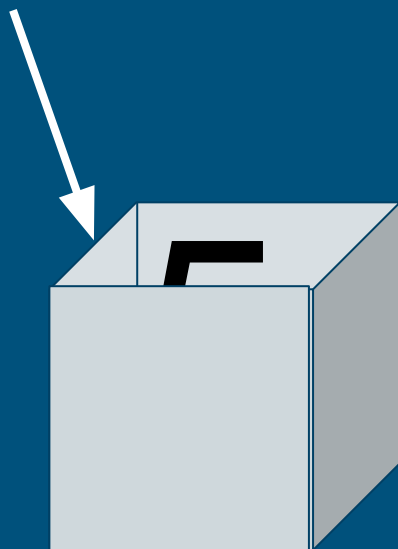


```
49 var to = "";  
50 to += persons[0] + " ";  
51 to += persons[1] + " ";  
52 to += persons[2] + " ";
```

ところでデータって何？

step05

変数  
(variable)



箱

変数の値  
(value)



5

中身

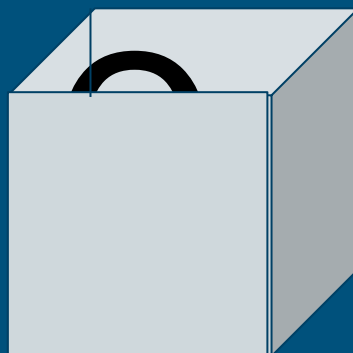
# 値が「ひとつだけ」の変数

```
var clickCount = 0;
```

変数(variable)

名前

```
var clickCount =
```



値(value)

# 変数 (variable) と値 (value)

```
var clickCount = 0;  
var persons = ['おじいちゃん', 'おばあちゃん', 'おばさん'];
```

変数(variable)の名前

変数(variable)

値(value)

var clickCount =

0

var persons =

おじいちゃん

おばちゃん

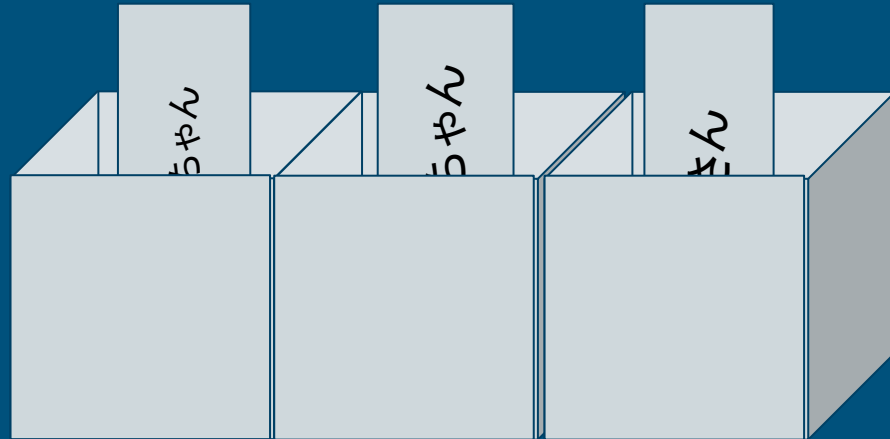
おばさん



# 値が「何個か」ある変数

```
var persons = ['おじいちゃん', 'おばあちゃん', 'おばさん'];
```

var persons =

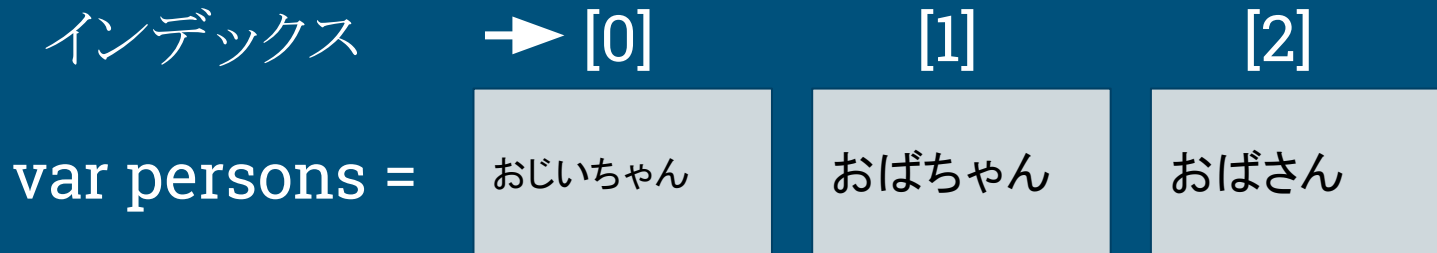


複数の値を複数の箱に入れる

配列 (array) と呼ぶ

# 配列 (array) とインデックス

それぞれの値を取り出すには...



それぞれの値を「要素(英語でelement)」と呼ぶ

要素を指すためにインデックスを使う

# バグ6。「おじさん」も足してあげよう！

## 「要素」を加える

```
3  var clickCount = 0;  
4  var persons = ['おじいちゃん', 'おばあちゃん', 'おばさん', 'おじさん'];
```

## 新しい要素を、インデックスを使ってメッセージに加える

```
49  var to = "";  
50  to += persons[0] + " ";  
51  to += persons[1] + " ";  
52  to += persons[2] + " ";  
53  to += persons[3] + " ";
```

# 配列の要素が増えるたびに書き足さないといけない→面倒くさい！

```
3  var clickCount = 0;  
4  var persons = ['おじいちゃん', 'おばあちゃん', 'おばさん', 'おじさん'];
```



増やす

```
49  var to = "";  
50  to += persons[0] + " ";  
51  to += persons[1] + " ";  
52  to += persons[2] + " ";  
53  to += persons[3] + " ";
```

← 新しい行を足す

もっと賢いやり方がある

バグ6。「おじさん」も足してあげよう！

```
49     var to = "";
50     to += persons[0] + " ";
51     to += persons[1] + " ";
52     to += persons[2] + " ";
53     to += persons[3] + " ";
```

「for ループ」をつかう。

それぞれの値を使って、同じ動作を繰り返す

```
50     var to = "";
51     for (var i = 0; i < persons.length; i++) {
52         to += persons[i] + " ";
53     }
```

# 完成！

メリークリスマス！



おじいちゃん おばあちゃん おばさん お  
じさん 楽しいクリスマスになりますよう  
に！