



String

文字の並びを表すデータ型

StringもJavaScriptのプリミティブ型の一つ。テキストを表現し、シングルクォート（'）やダブルクォート（"）で囲う必要がある。

STRING



```
let firstName = "Ziggy"; ダブルクォートで宣言  
let msg = "Please do not feed the chimps!";  
let animal = 'Dumbo Octopus'; シングルクォートもOK  
let bad = "this is wrong"; これはNG
```

シングルクォート、ダブルクォートのどちらを使っても大丈夫。
ただし、統一しておきましょう。

STRINGには インデックスがある

C	H	I	C	K	E	N
0	1	2	3	4	5	6

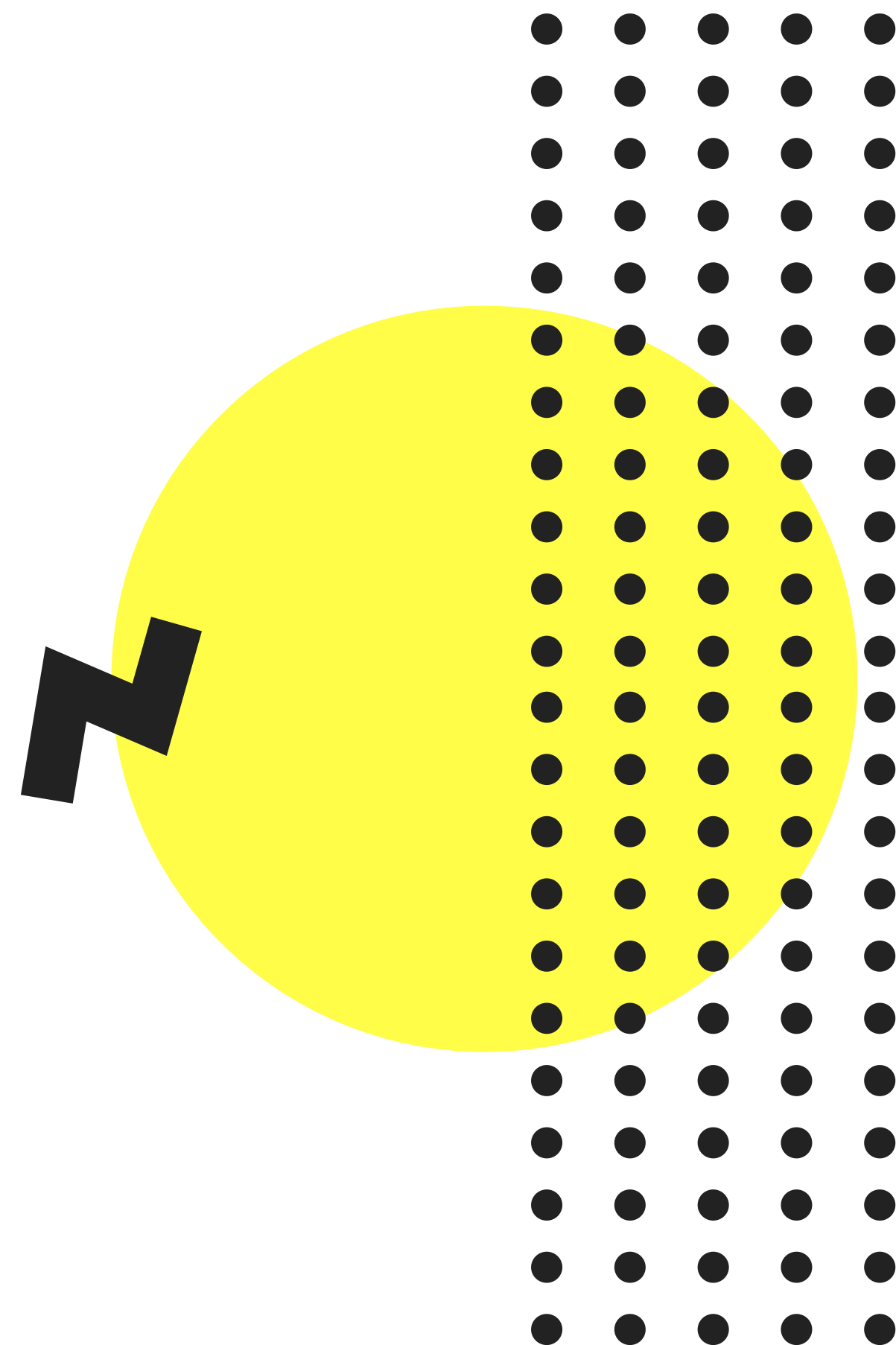
文字列の中の文字には添字でアクセスできる

Stringメソッド

組み込みの**String**メソッドで、文字列に対して便利な操作が実行できる

以下のようなことができる:

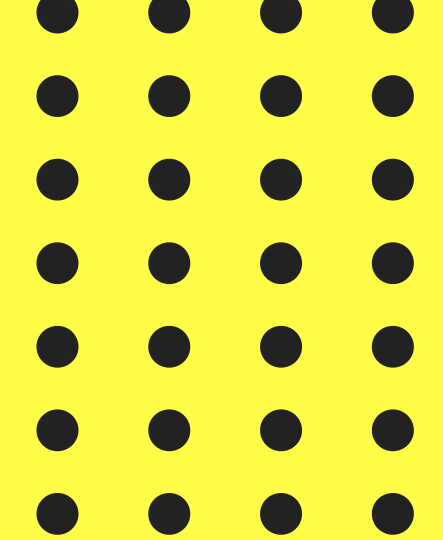
- 文字列内を検索できる
- 文字列内の文字を置換できる
- 大文字小文字を変換できる



thing.method()

casing

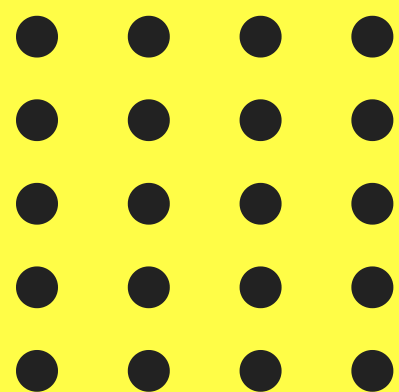
```
let msg = 'I am king';  
let yellMsg = msg.toUpperCase() // 'I AM KING'  
  
let angry = 'LeAvE mE aLoNe!';  
angry.toLowerCase(); // 'leave me alone!'  
  
// angryの中の値は変わらない  
angry; // 'LeAvE mE aLoNe!'
```



trim



```
let greeting = '  leave me alone plz  ';  
greeting.trim() // 'leave me alone plz'
```



thing.method(arg)

メソッドの括弧の中に**引数**を渡すこともできる。

引数を渡すことで動作するメソッドもある。

indexOf

```
let tvShow = 'catdog';  
  
tvShow.indexOf( 'cat' ); // 0  
tvShow.indexOf( 'dog' ); // 3  
tvShow.indexOf( 'z' ); // -1 (not found)
```

slice



```
let str = 'supercalifragilisticexpialidocious'  
  
str.slice(0,5); //'super'  
  
str.slice(5); // 'califragilisticexpialidocious'
```

replace



```
let msg = 'I am happy to be happy.';
```

```
msg.replace('happy', 'sad'); // 'I am sad to be happy.'  
// 1回目しか置換されないことに注意!
```

エスケープ表記

- \n – 改行
- \' – シングルクォート
- \" – ダブルクォート
- \\ – バックスラッシュ

テンプレートリテラル

超便利！

```
`I counted ${3 + 4} sheep`; // "I counted 7 sheep"
```

テンプレートリテラルは文字列の中に変数や式、改行を含めることができる便利な構文

使うのはバックティックス文字
(シングルクォートじゃない！)

``I am a template literal``

* 日本語キーボードなら Shiftキー + @

テンプレートリテラル



```
let item = 'cucumbers';  
let price = 1.99;  
let quantity = 4;  
  
`You bought ${quantity} ${item}, total price: ${price*quantity}`;  
// "You bought 4 cucumbers, total price: $7.96"
```

NULL & UNDEFINED

- Null
 - “意図的に値が存在しないことを表します”
 - 使うときは代入する
- Undefined
 - まだ値が代入されていない変数はundefined型となる

NULL

```
1 // ユーザーのログイン前
2 let loggedInUser = null; // 意図的に存在しないことを明示
3
4 // ユーザーがログインした
5 loggedInUser = 'Alan Rickman';
```

Undefined



```
1 let pickles; // 値を代入していない
2 pickles; // undefined
3 pickles = 'are very gross';
4
5 // undefinedは様々な状況で起きうる
6 let food = 'tacos';
7 food[7]; // undefined
```

MATH OBJECT

数学的な定数と関数を提供
するプロパティとメソッド
を持つ、組み込みのオブジ
ェクト

```
Math.PI // 3.141592653589793

// 四捨五入
Math.round(4.9) // 5

// 絶対値
Math.abs(-456) // 456

// 2の5乗
Math.pow(2,5) // 32

// 小数点以下を切り捨て
Math.floor(3.9999) // 3
```

RANDOM NUMBERS

Math.random()は0以上1未満のランダムな数字を返す



```
Math.random();  
//0.14502435424141957  
Math.random();  
//0.8937425043112937  
Math.random();  
//0.9759952148727442
```

ランダムな 数

1から10の間の整数を
生成してみよう

```
const step1 = Math.random();  
//0.5961104892810127  
const step2 = step1 * 10  
//5.961104892810127  
const step3 = Math.floor(step2);  
//5  
const step4 = step3 + 1;  
//6  
  
Math.floor(Math.random() * 10) + 1;
```