

# Javascript 入門

テキスト、数字と日付の処理

鄭祥飛@2021/04/28

# テキスト(文字列)の作成

以下の3つ方法があります。

```
let a = "hello world";  
let b = 'hello world'; // 効果は二重引用符と同じです  
let c = `hello world`;
```

# 文字列に改行を挿入

```
let a = "hello \n world";  
let b = 'hello \n world';  
let c = `hello  
world`;
```

# エスケープ文字

```
\b // バックスペース、前の位置もう時を削除
\f // 改ページ
\n // 改行
\r // Retrun
\t // Tabキー
\\ // \を入力する
\' // シングルクォーテーション
\" // ダブルクォーテーション
\xXX // 2桁の16進数が表すLatin-1文字
\uXXXX // 4桁の16進数が表すUnicode文字
\u{XXXXXX} // 16進数のコードポイントが表すUnicode文字
```

# \uXXXX と \u{xxxx}の区別

- \u{xxxx}:{}中に任意桁数の16進数
- \uXXXX : 一文字で表現できない場合、2つ\uXXXX組み合わせ或いはu{xxxx}で表現できる

例:

```
console.log("\uD867\uDE3D"); // => "鯨"
console.log("\u{29e3d}"); // => "鯨"
```

<https://jsprimer.net/basic/string-unicode/>

# 文字列に変数を挿入

```
let name = "□□□";  
let b = 'hello ' + name;  
let c = `hello ${b}`;  
let d = 'hello ${b}';
```

# 文字列長さ -- length

- エスケープ文字は一文字で計算する。
- バイト数で計算する

```
let name = "□□□";  
console.log(name.length);  
console.log("鮪".length);  
console.log("あ".length);  
const codePoints = Array.from("鮪");  
console.log(codePoints.length);
```

# 指定位置位置の文字を返す -- charAt

```
let msg2 = 'あいう鯨ういあ';
console.log(msg2.charAt(2)); // う
console.log(msg2.charAt(3)); // 鯨
console.log(msg2.charAt(4)); // 鯨
console.log(msg2.charAt(5)); // う
console.log("-----");
console.log(msg2.charCodeAt(2)); // うの代表する16進数を10進数に変換した数字
console.log(msg2.charCodeAt(3)); // 鯨の代表する16進数前半を10進数に変換した数字
console.log(msg2.charCodeAt(4)); // 鯨の代表する16進数後半を10進数に変換した数字
console.log(msg2.charCodeAt(5)); // うの代表する16進数を10進数に変換した数字
console.log("-----");
console.log(msg2.codePointAt(2));
console.log(msg2.codePointAt(3)); // 鯨の代表する16進数を10進数に変換した数字
console.log(msg2.codePointAt(4)); // 鯨の代表する16進数後半を10進数に変換した数字
console.log(msg2.codePointAt(5));
```



# 文字の所在位置を返す -- indexOf

```
let msg2 = 'あいう鮪ういあ';  
console.log("-----");  
console.log(msg2.indexOf("う"));  
console.log(msg2.lastIndexOf("う"));  
console.log(msg2.indexOf("いう"));  
console.log(msg2.indexOf("うい"));  
console.log(msg2.lastIndexOf("いう"));  
console.log(msg2.indexOf("鮪"));  
console.log(msg2.lastIndexOf("鮪"));  
console.log(msg2.indexOf("鮪あ"));
```

# 文字列検索

- `startsWith`
- `endsWith`
- `includes`

# 文字列分割と結合

- `concat` : 文字列の結語
- `split` : 文字列を部分文字列へと分けることで、String オブジェクトを文字列の配列に分割します。
- `slice(beginIndex, endIndex)` : 文字列の一部を取り出し、新しい文字列を返します。
  - `beginIndex` と `endIndex` はマイナス指定できる。例: `-2 = str.length-2`
  - `endIndex` を省略した場合、`slice()` は文字列の末尾までを取り出します

# 文字列の一部の抽出

- `substring(indexStart, indexEnd)`
  - `indexEnd`: 関数文字は、`indexEnd`の文字は含めない。省略場合、最後まで。
  - `indexStart:indexEnd` より大きかった場合、2つの引数が交換されたものとして実行されます。
- `substr(start[, length])`
  - `start`: 正数場合、文字列の先頭から数えた位置になります。
  - `start`: 負数場合、文字列の末尾から数えた位置になります。
  - `length`: 省略された場合、`substr()` は文字列の末尾までの文字を抽出します。

# 大文字・小文字变换

- toLowerCase
- toUpperCase

# 空白の削除

- `trim()`
- `trimLeft()`
- `trimRight()`

# 宿題

1. 以下の文字列中に、各単語の先頭文字を大文字に変更し、余りの文字を小文字に変更する

```
Split pages by horizontal ruler (`---`). It's very simple! :satisfied:
```

2. 以下の文字列中の各単語から「o」以下の文字を出力する。

```
If you know how to write document with Markdown, you already know how to  
write Marp slide deck too.
```

出力: ou,ow,ow,ocument...

# 数値の種類

- 10進数: 1234567890 //0777 8 進数として解析される
- 8進数: 0o10
- 2進数: 0b10000
- 16進数: 0x123456789ABCDEF



# BigIntと指数

- `BigInt: 9007199254740991n ; BigInt(9007199254740991);`
- `1e3: //1000`

# 数字に関する処理プロパティと関数

- Number.MAX\_VALUE
- Number.MIN\_VALUE
- Number.parseFloat()
- Number.parseInt()
- toFixed():数値の丸め処理。四捨五入
- toPrecision():指定された精度で表した文字列を返します。  
12.1233.toPrecision(5):12.123

# toFixed()の不思議の所

- `12345.6789.toFixed()` // `'12346'` を返す :
- `12345.6789.toFixed(1)` // `'12345.7'` を返す :
- `2.35.toFixed(1)` // `'2.4'` を返す。切り上げ。
- `2.55.toFixed(1)` // `'2.5'` を返す。切り捨て。

# toFixd()の強化

資料を調べてみてください!

# 数学関数

- `abs()` : 絶対値
- `sin()`, `cos()`, `tan()`: 三角関数
- `Math.floor()`: 与えられた数値以下の最大の整数を返します
- `Math.random()`: 0 以上 1 未満 (0 は含むが、1 は含まない) の範囲で浮動小数点
- `Math.round()`: 与えた数を四捨五入して、もっとも近似の整数を返します。
- `Math.Max()`
- `Math.min()`

# 宿題

5つ100以下の乱数を生成して、配列に設定する。その中の最大値を見つけて出力する。

# 日付対象の作成

- `new Date()`
- `new Date(1999,11,25)`
- `new Date("2011-11-12")`

# 関数

- `getFullYear()`;
- `setFullYear()`;
- `getTime()`: ミリ秒
- `getMonth()`: 0から
- `getDay()`: 週
- `getDate()`: 月のなん日
- `getTimezoneOffset()`: 世界協定時と時差。分



# 関数

- `toDateString()`
- `toTimeString()`
- `toLocaleDateString()`
- `toLocaleTimeString()`
- `toUTCString()`

# 宿題

- 今の時間を「yyyy/MM/dd HH:mm:ss」フォーマットを出力する
- 年末まで、残日数
- 下記の人から、今日誕生日人の名前を出力する。
  - 王さん： 1998-12-12、
  - 周さん： 2000-3-1、
  - 張さん： 1 9 9 1 — 1 0 — 1 1 、
  - Aさん： 1990-05-12,
  - Bさん： 2009-5-13