

# JavaScript

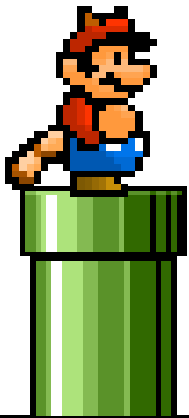
## プログラミング入門

### 3-3 Stringオブジェクトの文字列操作のメソッド

3-3-1 指定した位置の文字列を取り出す

3-3-2 文字列の位置を調べる

3-3-3 Stringオブジェクトのその他のメソッド



2006年10月27日(金) 南 慶典

## 3-3-1 指定した位置の文字列を取り出す

Stringオブジェクトに格納されている文字列の中から、指定した範囲の文字を取り出すメソッドを紹介。

- **charAt()メソッド**  
引数で指定した位置の1文字を取り出す
- **substring()メソッド**  
Stringオブジェクトから指定した範囲の文字列を取り出す
- **slice()メソッド**  
指定した範囲の文字列を取り出す。ただし、2番目の引数に負の値を指定した場合には、文字列は最後から数える。
- **substr()メソッド**  
指定した範囲の文字列を取り出す。ただし、2番目の引数には取り出す文字の個数を指定する。

# charAt()メソッド ①

- **charAt()メソッド**

引数で指定した位置の1文字を取り出す

書式:

▶ Stringオブジェクト名.charAt(index)

引数:

▶ index —— 最初の文字を0とする文字の位置

戻り値:

▶ 引数indexで指定した位置の文字(Stringオブジェクト)

注意:

- 引数は先頭を0とする。配列の添え字と同じ。

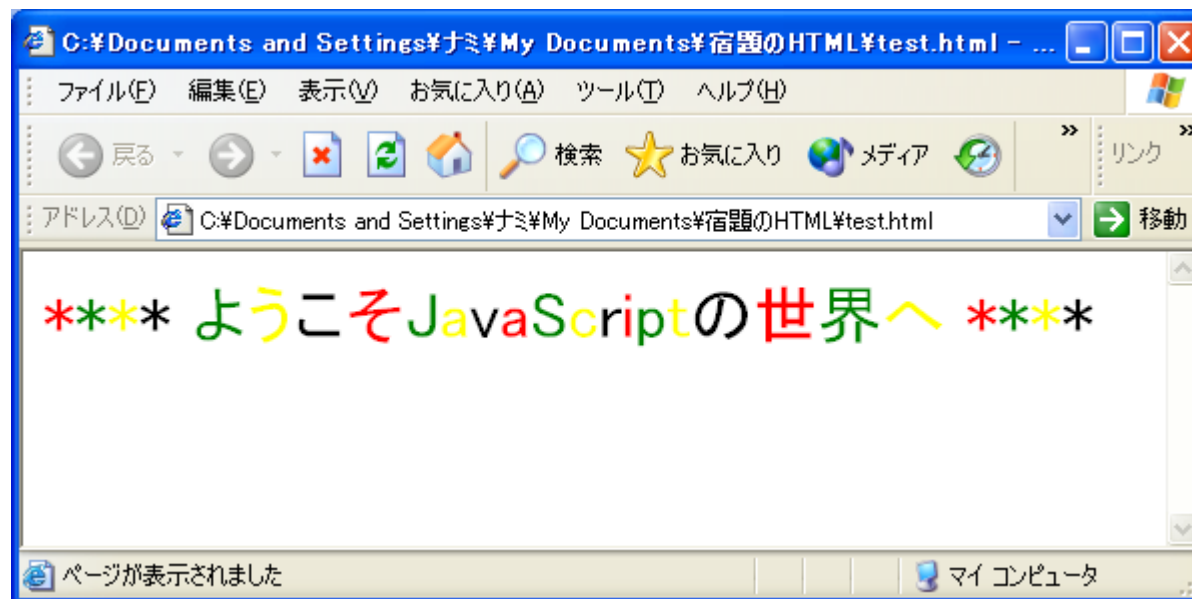
# charAt()メソッドの使用例

```
</head>
<body bgcolor="#ffffff">
<div style="font-size:24pt">
<script language="Javascript" type="text/javascript">
var myString = "**** ようこそJavaScriptの世界へ ****"

// 色の配列を作成
var cArray = new Array("red","green","yellow","black");
var cNum = 0;

for(var i=0;i<myString.length;i++){
    var tempStr = myString.charAt(i);
    cNum = i % cArray.length;
    document.write(tempStr.fontcolor(cArray[cNum]));
}
</script></div>
▪
▪
```

# 実行結果



# substring()メソッド ①

## ● **substring()メソッド**

Stringオブジェクトから指定した範囲の文字列を取り出す

書式:

▶ Stringオブジェクト名.substring(index1, index2)

引数:

▶ index1 —— 取り出す文字列の先頭の位置

▶ index2 —— ... の最後の文字の次の位置

戻り値:

▶ index1からindex2までの文字列(Stringオブジェクト)

注意:

- 引数は先頭を0とする。配列の添え字と同じ。
- index1とindex2を同じ値を指定した場合、空文字列が返される。
- 2番目の引数を省略した場合、文字列の最後まで返される。

## substring()メソッド ②

- 変数myStringに文字列"ABCDEFGH"が格納されている場合

実行例

戻り値

myString.substring(0, 2)

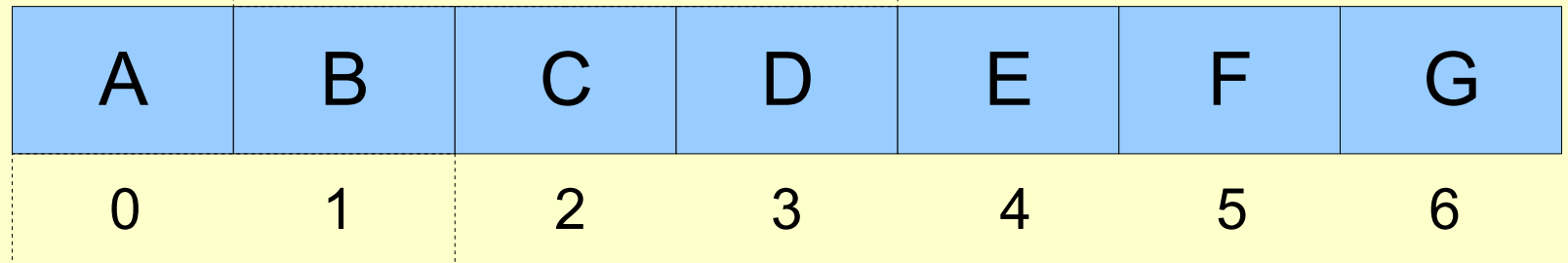
"AB"

myString.substring(1, 4)

"BCD"

myString.substring(1, 4)

変数myString



myString.substring(0, 2)

# slice()メソッド ①

## ● slice()メソッド

指定した範囲の文字列を取り出す。ただし、2番目の引数に負の値を指定した場合には、文字列は最後から数える。

書式:

▶ Stringオブジェクト名.slice(index1,index2)

引数:

▶ index1 —— 取り出す文字列の先頭の位置

▶ index2 —— ・・の最後の文字を後ろから数えた位置

戻り値:

▶ index1からindex2までの文字列(Stringオブジェクト)

注意:

- 引数は先頭を0とする。後ろからの数え方は0, -1, -2 ...
- 2番目の引数を省略した場合、文字列の最後まで返される。

## slice()メソッド ②

- 変数myStringに文字列"ABCDEFGG"が格納されている場合

実行例

戻り値

myString.slice(0, -1)

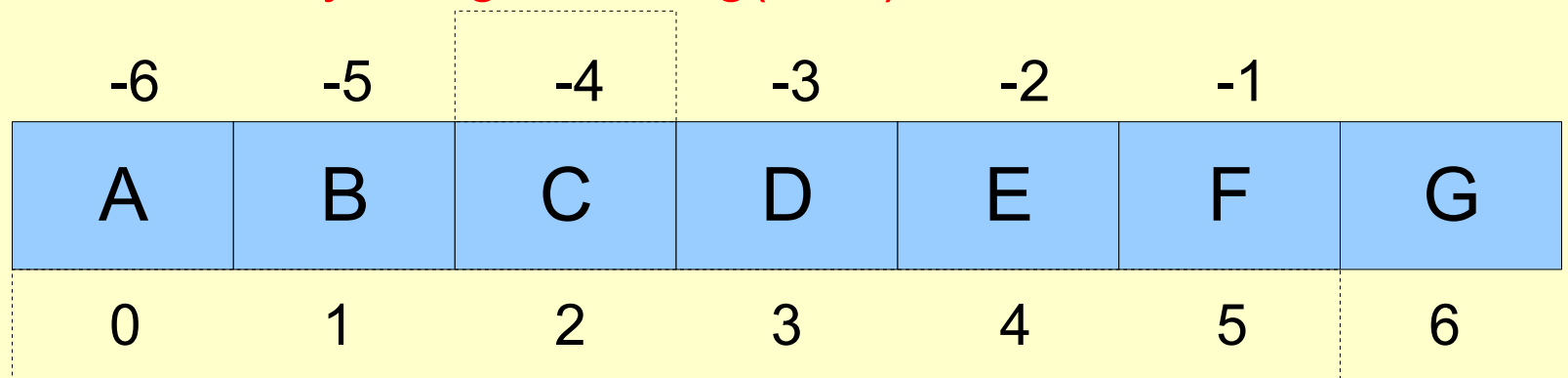
"ABCDEF"

myString.slice(2, -4)

"C"

myString.substring(2, -4)

変数myString



myString.slice(0, -1)

# substr()メソッド

## ● substr()メソッド

指定した範囲の文字列を取り出す。ただし、2番目の引数には取り出す文字の個数を指定する。

書式:

▶ Stringオブジェクト名.substr(index1, num)

引数:

▶ index1 —— 取り出す文字列の先頭の位置

▶ index2 —— 取り出す文字の個数

戻り値:

▶ index1からnum個までの文字列(Stringオブジェクト)

注意:

- 2番目の引数を省略した場合、文字列の最後まで返される。
- numが0の場合、空文字が返される。

## 3-3-2 文字列の位置を調べる

- **indexOf()メソッド**  
指定した文字列が、Stringオブジェクト内に含まれていた場合、先頭からの位置を戻す
- **lastIndexOf()メソッド**  
StringIndexOfとは逆に、指定した文字列が最後に見つかった位置を戻す

# indexOf()メソッド ①

## ● **indexOf()メソッド**

指定した文字列が、Stringオブジェクト内に含まれていた場合、先頭からの位置を返す

書式:

- ▶ Stringオブジェクト名.indexOf(SearchValue)
- ▶ Stringオブジェクト名.indexOf(SearchValue, fromIndex)

引数:

- ▶ SearchValue —— 調べる文字列
- ▶ fromIndex —— 文字列を検索する開始位置

戻り値:

- ▶ SearchValueが見つかった位置を表す数値

注意:

- 先頭の文字は0、見つからなかった場合は-1を返す

## indexOf()メソッド ②

- 変数myStringに文字列"ABCDEFGH"が格納されている場合

実行例

戻り値

myString.indexOf("BC")

1

myString.indexOf("E", 3)

4

myString.indexOf("e", 3)

-1

myString.indexOf("CD", 2)

2

myString.indexOf("CD", 3)

-1

# lastIndexOf()メソッド ①

## ● lastIndexOf()メソッド

StringIndexOfとは逆に、指定した文字列が最後に見つかった位置を戻す

書式:

- ▶ Stringオブジェクト名.lastIndexOf(SearchValue)
- ▶ Stringオブジェクト名.lastIndexOf(SearchValue, fromIndex)

引数:

- ▶ SearchValue —— 調べる文字列
- ▶ fromIndex —— 文字列を検索する開始位置

戻り値:

- ▶ SearchValueが見つかった位置を表す数値

注意:

- 先頭の文字は0、見つからなかった場合は-1を返す

## lastIndexOf()メソッド ②

- 変数myStringに文字列"ABCABC"が格納されている場合

実行例

戻り値

myString.lastIndexOf("BC")

4

myString.lastIndexOf("E")

-1

myString.lastIndexOf("BC", 3)

1

myString.lastIndexOf("ABC", 1)

3

# lastIndexOf()メソッドの使用例

getFileName.html (URLからファイル名の部分だけを取り出して表示)

```
var URL = location.href;  
var fileName = URL.substring(URL.lastIndexOf("/") + 1, URL.length);  
document.write("<p>", fileName, "</p>");
```

- ▶ 「location.href」は現在ロードされているHTMLドキュメントのURLを格納するプロパティ
- ▶ URL.lastIndexOf("/")で最後の「/」の位置を見つけ、substring()メソッドでそれ以降の文字列を取り出す
- ▶ 変数fileNameには、「getFileName.html」が格納される

## 3-3-3 Stringオブジェクトのその他のメソッド

- **toLowerCase()メソッド**  
オブジェクト内の半角・全角の大文字アルファベットを小文字にする
- **toUpperCase()メソッド**  
オブジェクト内の半角・全角の小文字アルファベットを大文字にする
- **split()メソッド**  
オブジェクト内の文字列を配列に変換する
- **charCodeAt()メソッド**  
指定した位置の文字の文字コードを調べる
- **fromCharCode()メソッド**  
文字コードを文字に変換する

# split()メソッド

## ● split()メソッド

オブジェクト内の文字列を配列に変換する

書式:

- ▶ Stringオブジェクト名.split()
- ▶ Stringオブジェクト名.split(separator)

引数:

- ▶ Separator —— セパレータとして使用する文字

戻り値:

- ▶ Stringオブジェクトの配列

注意:

- separatorを指定しない場合、オブジェクト全体を一つの要素とする配列を戻す

# split()メソッドの使用例

split.html (split()メソッドの使用例)

```
var myString = "Rock,Jazz,Pops";  
var myArray = myString.split(",");  
for(var i=0;i<myArray.length;i++){  
    document.write(myArray[i],"<br>");  
}
```

● 配列 myArray

myArrayの要素

値

myArray[0]

Rock

myArray[1]

Jazz

myArray[2]

Pops

# charCodeAt()メソッド

- **charCodeAt()メソッド**  
指定した位置の文字の文字コードを調べる

書式:

▶ Stringオブジェクト名.charCodeAt(index)

引数:

▶ index —— 文字の位置

戻り値:

▶ 文字コード

注意:

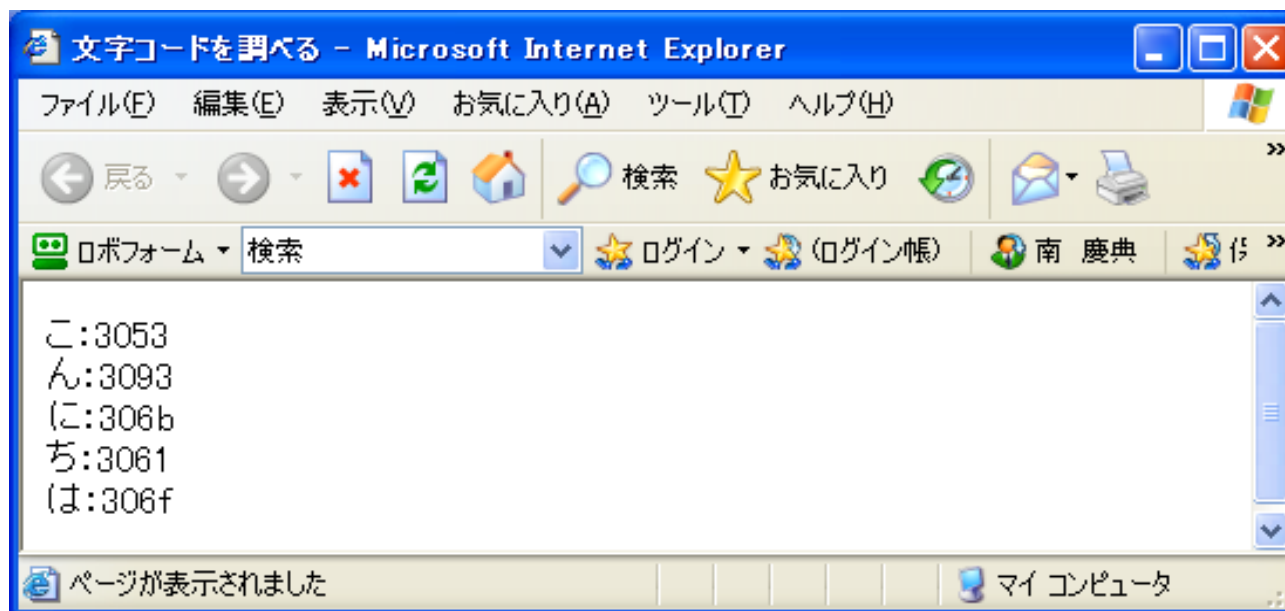
- 文字コードは、文字がASCIIキャラクタの場合にはASCIIコードで、日本語など2バイト文字の場合にはUnicodeが戻される

# charCodeAt()メソッドの使用例

charCodeAt.html(スクリプト部分) (先頭から順に文字と文字コードを表示)

```
var str = "こんにちは";  
for(var i=0; i < str.length; i++){  
    document.write(str.charAt(i),":",  
                    str.charCodeAt(i).toString(16), "<br>");  
}
```

- 変数strに「こんにちは」を格納した場合の実行結果



# fromCharCode()メソッド

- **fromCharCode()メソッド**  
文字コードを文字に変換する

書式:

▶ `String.fromCharCode(code1, code2, ...)`

引数:

▶ `codeN` —— 文字コード

戻り値:

▶ Stringオブジェクト

注意:

- このメソッドはstaticメソッドなので、インスタンスを作らなくても呼び出せる

# fromCharCode()メソッドの使用例

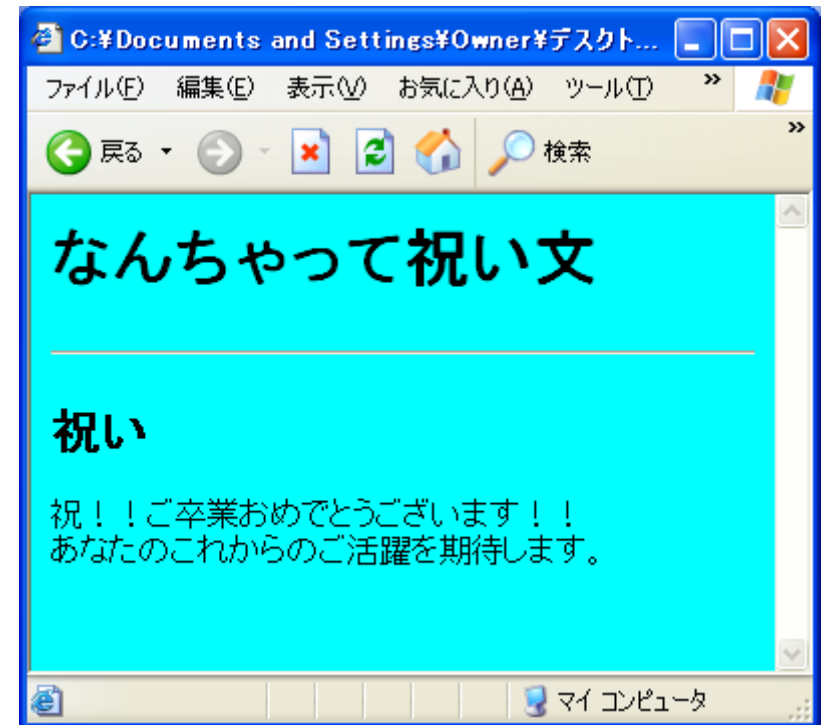
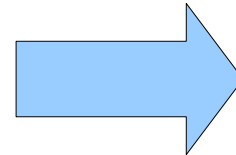
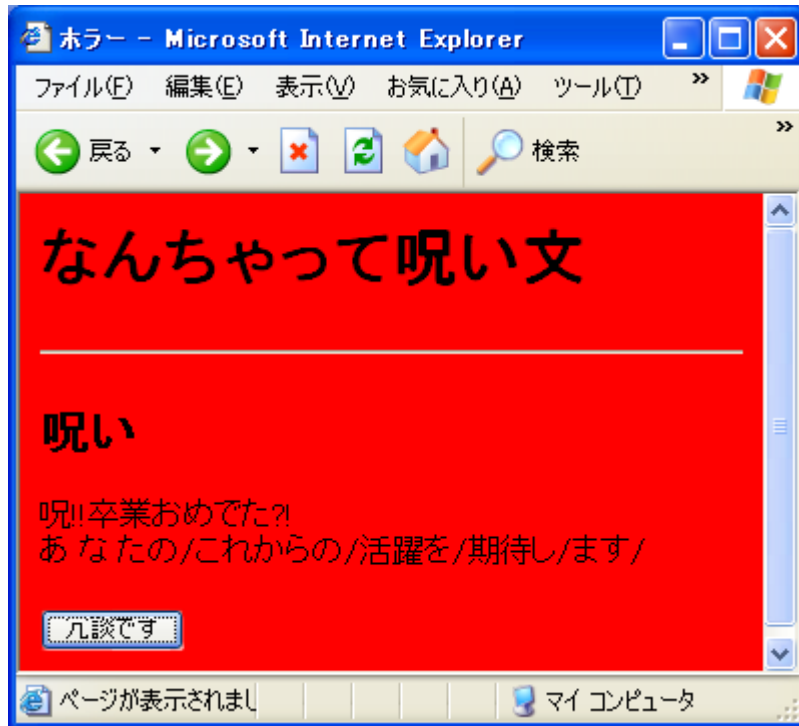
fromCharCode.html(スクリプト部分) (文字コードから「こんにちは」)

```
var str ;  
str = String.fromCharCode(0x3053, 0x3093, 0x306b,  
                           0x3061, 0x306f);  
document.write(str);
```

- 文字コードから文字列「こんにちは」を表示



# 宿題



「なんちゃって祝い文」を右図の「なんちゃって祝い文」に変えなさい。

ただし、..

- ・全ての文字列は変数に入れて表示させること
- ・文字列を違う文字列に変えたい場合は、紹介したメソッドを使うこと