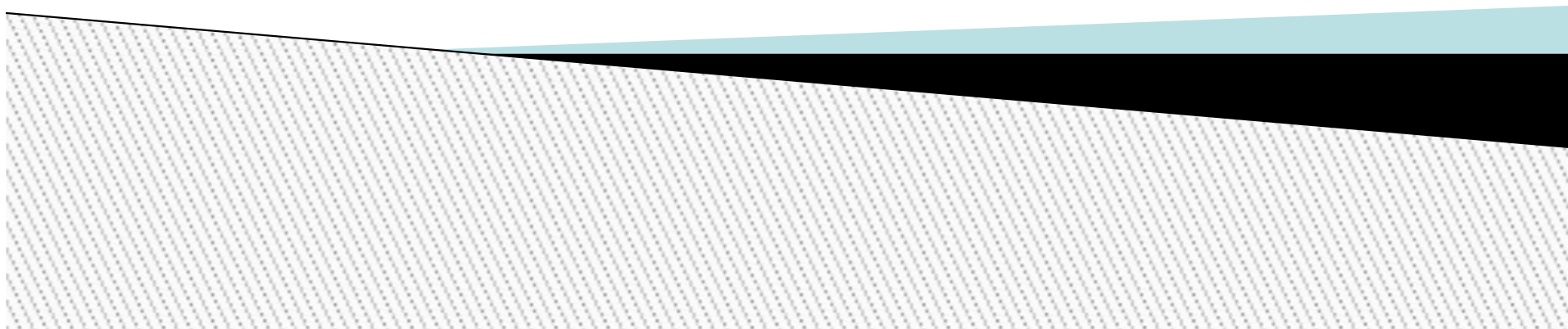


JavaScript入門

3.1~3.3

08t4072| 全太俊



3-1-1 Stringオブジェクトの生成

- 文字列を「`""`」もしくは「`'`」で囲む

```
var studentName="Makoto ohtsu";  
var teacherName="Taro Yamada";
```

- Stringコンストラクタを使用する

```
var studentName=String"Makoto ohtsu";  
var teacherName=String"Taro Yamada";
```

3-1-2 Stringオブジェクトの結合

+演算子を使用すると結合することができる

- 文字列どうしの結合

```
var firstName = "Makoto";
```

```
var LastName = "Otsu";
```

```
var fuLLName = firstName + "" + lastName;
```

➤ fuLLNameの値は「Makoto Otsu」

- 文字列と数値の結合

```
var empLoyeeNum = 101;
```

```
var empLoyeeStr = "EMPLOYEE:" + empLoyeeNum
```

➤ empLoyeeStr の値は「EMPLOYEE:101」

Stringオブジェクトの結合

- 数値をStringオブジェクトに変換する

数字と空文字列「」を連結

```
var myName = 10;  
var myNumStr = "" + myNum;
```

toStringメソッド

書式: 変数.toString()

変数.toString(radix)

引数: radix (基数を示す数値)

戻り値: Stringオブジェクト

```
var num = 10;  
var numStr = num.toString();文字列「10」  
var num = 10;  
var numStr = num.toString(2);2進数に変換「1010」  
var num = 255;  
var numStr = num.toString(16);16進数に変換「ff」
```

3-1-3 Stringオブジェクトのlengthプロパティ

- 文字列の長さは、lengthプロパティに入れている

```
var studentName = "taro";
```

```
var nameLength = studentName.length;
```

➤ nameLengthの値は「4」

- テキストフィールドの文字数のチェック

ソースは省略

onchangeイベントハンドラ

3-1-4 Stringオブジェクトのメソッドの取り扱い

- toUpperCase()メソッド: 文字列を大文字に変換

```
var myString="abc";
```

```
var myUpperString=myString.toUpperCase();
```

注意: 半角英小文字に対して有効

- bold()メソッド: ボールド体にする

```
var myString="makoto otsu";
```

```
var myUpperString=myString.toUpperCase();
```

3-2-1 ハイパーリンクを設定する

- アンカーを設定する

メソッド: `anchor()`

書式: `String`オブジェクト名.`anchor(anchorName)`

引数: `anchorName` – アンカー名

戻り値: アンカーとなった`String`オブジェクト

```
var myString = “目次”;
```

```
document.write(myString.anchor(“index”));
```

`目次`と同じ

- リンクを設定する

書式: `String`オブジェクト名.`link(href)`

引数: `href` – ハイパーテキストの移動先 (URL または アンカー)

戻り値: リンクとなった`String`オブジェクト

```
var toOtm = “オーム社”;
```

```
document.write(toOtm.Link(“http://www.ibaraki.ac.jp”));
```

3-2-2 文字の大きさを設定する

- 文字を大きく表示

メソッド: `big()`

書式: Stringオブジェクト名.`big()`

戻り値: フォントが大きくなったStringオブジェクト

- 文字を小さく表示

メソッド: `small()`

書式: Stringオブジェクト名.`small()`

戻り値: フォントが小さくなったStringオブジェクト

- 文字の大きさを設定する

Stringオブジェクトメソッド (size)

引数: size—フォントのサイズ

戻り値: フォントサイズの変更されたStringオブジェクト

3-2-3 書体を設定する

- 書体を設定するメソッド

メソッド名

bold()

italics()

strike()

sup()

sub()

fixed()

説明

書体を太文字(ホールド)にする

書体を斜体(イタリック)にする

文字列を抹消線付きで表示する

表示される文字列を上付き文字にする

表示される文字列を下付き文字にする

文字列のフォントを設定されWebブラウザで等幅

3-2-4 文字の色を変更する

- 文字列の色を変える

メソッド: `fontcolor()`

書体: `String`オブジェクト名.`fontcolor(color)`

引数: `color`—色の名前またはRGB値 (`#RRGGBB`)

戻り値: 色が変更された`String`オブジェクト

3-3-1 指定した位置の文字列を取り出す

- `charAt()`メソッド: 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)
- `substring()`メソッド: 指定した文字列を取り出す
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`——取り出す文字列の最後の位置
戻り値: `index1`から`index2`までの文字列(`String`オブジェクト)
`myString="ABCDEFGG";`

実行例	戻り値
<code>myString.substring(0,2)</code>	"AB"
<code>myString.substring(1,4)</code>	"BCD"

メソッド

- slice()メソッド: index2が負のときは後ろから数える
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 — 取り出す文字列の最後の位置

実行例	戻り値
myString.substring(0,-1)	“ABCDEF”
myString.substring(2,-4)	“C”

- substr()メソッド
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

実行例 m	戻り値
myString.substr(3,3)	“DEF”
myString.substr(2,2)	“CD”

3-3-2 文字列の位置を調べる

- indexOf()メソッド

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

- lastIndexOf()メソッド

書式: Stringオブジェクト.LastIndexOf(searchValue)

Stringオブジェクト.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

3-3-3 Stringオブジェクトのその他のメソッド

- toLowerCase()メソッド

書式: Stringオブジェクト名.toLowerCase()

戻り値: 小文字に変換されたStringオブジェクト

- toUpperCase()メソッド

書式: Stringオブジェクト名.toUpperCase()

戻り値: 大文字に変換されたStringオブジェクト

- split()メソッド

書式: Stringオブジェクト.split()

Stringオブジェクト.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

メソッド

- ・ `charCodeAt()`メソッド

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

- ・ `fromCharCode()`メソッド

書式: `String.fromCharCode(code1,code2,....)`

引数: `codeN`—文字コード

戻り値: `String`オブジェクト

宿題

赤、青、緑のボタンを作り、ボタンで背景の色を変えるプログラムを作ってください