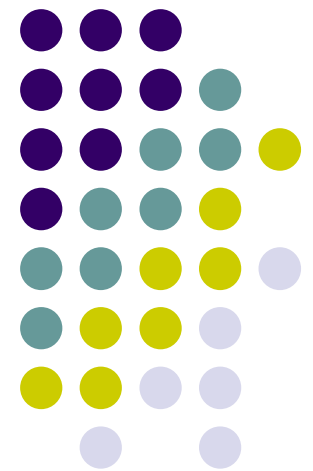


# 第7回 JavaScriptゼミ

---

## 4 - 3 documentオブジェクト

発表日 : 06/11/24 発表者 : 鈴木朋央



# documentオブジェクトの概要



windowオブジェクトの下の階層に位置するため、windowオブジェクトのプロパティとしてアクセス可能

```
windowリファレンス. document .メソッド名()
```

```
windowリファレンス. document .プロパティ名
```

windowリファレンス : HTMLドキュメントが属するwindowオブジェクトへの参照

例 : ドキュメントがロードされているウィンドウ自体のwindowリファレンス

「window」 もしくは 「self」

# 色を表すプロパティ



## alinkcolor

アクティブリンク (リンクをクリックした瞬間からデータを読み込んでいる間) の色。デフォルトは<body>タグのalinkアトリビュートの値

## linkcolor

リンク (<a href></a>タグの部分) のデフォルトの色。デフォルトは<body>タグのlinkアトリビュートの値

## vlinkcolor

訪問済みのリンクの色。デフォルトは<body>タグのvlinkアトリビュートの値

## bgcolor

ウィンドウの背景色。デフォルトは<body>タグのbgcolorアトリビュートの値

## fgcolor

テキストのデフォルトの色。デフォルトは<body>タグのtextアトリビュートの値



# プロパティの設定方法

色を表すプロパティに値を設定する場合は、#RRGGBBを用いる

```
document.linkColor = "#808000";
```

もしくは、色の名前で指定

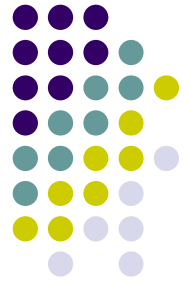
```
document.bgColor = "Olive";
```

以下の場合、変数の値が代入されてしまう

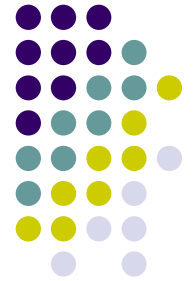
```
document.bgColor = Olive;
```

\* 色の名前は大文字、小文字は区別されない

# HTMLドキュメントの情報を保持するプロパティ



プロパティ名	説明
title	HTMLドキュメントのタイトル。HTMLのtitleエレメントで設定されている文字列
lastModified	HTMLドキュメントの最終更新日
referrer	HTMLドキュメントを呼び出したページのURL
URL	HTMLドキュメントのURL
domain	URLのドメイン部分(ローカルファイルから取得した場合には値はない)



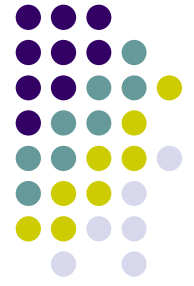
# プロパティへのアクセス例

```
function showInfo(){  
    var infoStr;  
    infoStr = "\nタイトル:" + document.title + "\n";  
    infoStr += "最終更新日:" + document.lastModified + "\n";  
    if (document.referrer != ""){  
        infoStr += "呼び出しページ:" + document.referrer + "\n";  
    }  
    infoStr += "URL:" + document.URL;  
    window.alert(infoStr);  
}
```

実行結果

情報を開示する





# プロパティを配列としてアクセス

documentオブジェクトは、formやlink、anchorオブジェクトなどのHTMLの要素をプロパティとして保持している

要素にアクセスするには…

- ・ HTMLの各タグのnameアトリビュートで設定した名前
- ・ 配列として

配列名	説明
forms	documentオブジェクト内のすべてのformオブジェクトを格納する配列
links	documentオブジェクトのすべてのlinkオブジェクトを格納する配列
anchors	documentオブジェクト内のすべてのanchorオブジェクトを格納する配列
images	documentオブジェクト内の、Imageオブジェクトの配列 (HTMLのimgエレメントによって生成されたもの)



# アクセス例

配列の添字は、HTMLドキュメント内に記述されている順番となる

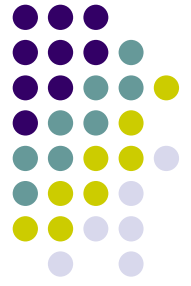
例：フォームが3つ記述されているとき

- ・ 最初のフォーム `document.forms[0]`
- ・ 2番目のフォーム `document.forms[1]`
- ・ 3番目のフォーム `document.forms[2]`

例：3つのフォームの名前をそれぞれF1,F2,F3としたとき

- ・ 最初のフォーム `document.F1`
- ・ 2番目のフォーム `document.F2`
- ・ 3番目のフォーム `document.F3`

# cookieの概要

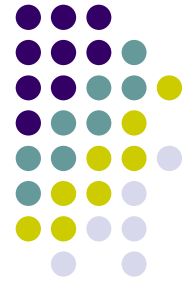


クライアントサイドJavaScriptでは、セキュリティ上の配慮からローカル、リモート問わずファイルの読み書きが禁止されている

ただし、例外として「**cookieファイル**」と呼ばれるファイルだけは例外であり、JavaScriptからcookieプロパティとしてアクセス可能

クライアント側のハードディスク内に置かれるテキストファイルで、ユーザ情報などを一時的に保持しておくことが可能

ファイルに格納しておく情報の単位を「cookie」と呼び、最大300までのcookieを格納可能(一つのcookieサイズは最大4KB)



# cookieの作成方法

**NAME=VALUE;expires=DATE**

NAME:cookieの名前

VALUE:格納する実際の値

(日本語の場合encodeURIComponent()関数でエンコードする)

expires=DATE:有効期限 (グリニッジ標準時形式で記述)

例:3日間有効なcookieを作成し、「大津真」という文字列を格納

```
var expire=new Date();
```

```
expire.setTime(expire.getTime() + (3*24*60*60*1000));
```

```
var dateStr = expire.toGMTString();
```

```
document.cookie = "@name" + encodeURIComponent("大津真") +  
";" + "expires" + dateStr;
```



# cookieの読み込み方法

読み出されたcookieは次のような形式で戻される

**NAME1=VALUE1;NAME2=VALUE2;NAME3...**

例:「@name」と「@place」2つのcookieが登録されていた場合

```
document.write(document.cookie, "<br>");
```

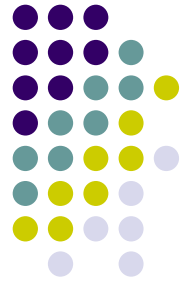
を実行してみると...

```
@name=%E5%A4%A7%E6;@place=TOKYO
```

と出力される

目的のデータを得るためには、cookieを切り分ける必要がある

# cookieの完全な書式



`NAME=VALUE;expire=DATE;path=PATH;domain=DOMAIN;secure`

## PATH

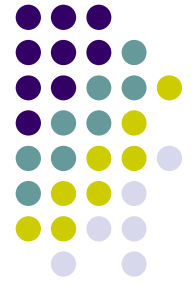
cookieの有効なパスを示す。省略された場合にはHTMLドキュメントがロードされたパスになる

## DOMAIN

cookieの有効なドメイン名を示す。省略された場合にはHTMLドキュメントがロードされたサーバのドメイン名になる

## secure

セキュリティの設定。値は「true」または「false」のどちらかで、デフォルトではfalseになる。trueに設定された場合には、HTTPS(HTTP over SSL)サーバなど安全性が保証されている場合のみ送信される



# その他のメソッド

メソッド: `writeln()`

構文: `document.writeln(expression1, ..... , expressionN)`

引数: `expression1, ..... , expressionN` ---- 出力する式、または文字列

説明: 引数として渡された式または文字列を、改行つきでドキュメントに出力する

\* HTMLドキュメントでは改行が無視されてしまうため、実際にWebブラウザのウィンドウで改行させるためには、`<pre></pre>`タグで囲む必要がある

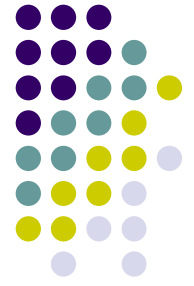
使用例:

```
document.write("<pre>");
```

```
document.writeln("message1");
```

```
document.writeln("message2");
```

```
document.write("</pre>");
```



# その他のメソッド

メソッド: `open()`

構文: `document.open(mimetype)`

引数: `mimeType` --- MIMEタイプ

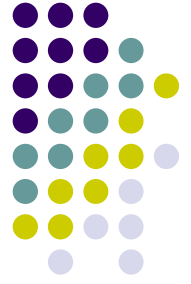
説明: 引数`mimeType`で指定した形式で出力ストリームを開く。`mimeType`を省略した場合には「`text/html`」つまりHTMLドキュメントと見なされる。現在のウィンドウに出力する場合には`open()`メソッドは必要ない。別のウィンドウを開いたような場合には、`open()`メソッドと、次に示す`close()`メソッドが必要になる。

メソッド: `close()`

構文: `document.close()`

説明: `close()`メソッドは、出力ストリームを閉じ実際に画面に書き出す

# 練習問題



付録のサンプルプログラムに、cookieを用いてのユーザ認証システムを追加してください