

第2回 JavaScriptゼミ

2 - 4 制御構造

発表者 田島勇樹

発表日 2006年10月13日(金)

2-4-1 if文(1)

- if文

条件によって処理を変える制御構造

書式

```
if (条件式) ステートメント;
```

条件式が真: ステートメントを実行

条件式が偽: ステートメントを実行しない

- if文の例

```
var i=3;
```

```
...
```

```
if (i < 0) i=0;
```

2-4-1 if文(2)

- 条件が成り立った場合に複数のステートメントを実行一連のステートメントを実行するには「{}」くくってブロック化する。

```
if (条件式){  
    ステートメント;  
    ステートメント;  
    .....  
}
```

- 条件に合わなかった場合の処理を指定する
else句を用いて条件式の結果が偽の場合の処理を記述する。

```
if (条件式)  
    ステートメント1;  
else  
    ステートメント2;
```

2-4-1 if文(3)

- if文を用いて2つの値を比較した例

```
var value1 = 5;  
var value2 = 4;  
if (value1 < value2)  
    result = value2;  
else  
    result = value1;  
document.write(result);
```



else句の場合も「{ }」を使えば一連のステートメントを実行できる。

```
if(条件式){  
    ...  
}  
else{  
    ステートメント1;  
    ステートメント2;  
    ...  
}
```

2-4-1 if文(4)

- 条件演算子

条件式の値によって対応する式の結果を返すための演算子

書式

条件式 ? 式A : 式B

- 条件演算子の例

```
result = (value1 > value2) ? value1 : value2;
```

もしvalue1が大きいときresultにvalue1の値が格納される。

もしvalue1が小さいときresultにvalue2の値が格納される。

JavaScriptでは、条件式の結果が0,null,undefined,および空文字列「」以外は全て「真」と判断する

2-4-2 for文(1)

- for文

制御変数を用意し、ある条件式が真になるまで一連のステートメントを繰り返し実行する。

書式

```
for (初期処理式; 条件式; 再処理式) ステートメント;
```

- ブロック化する場合

書式

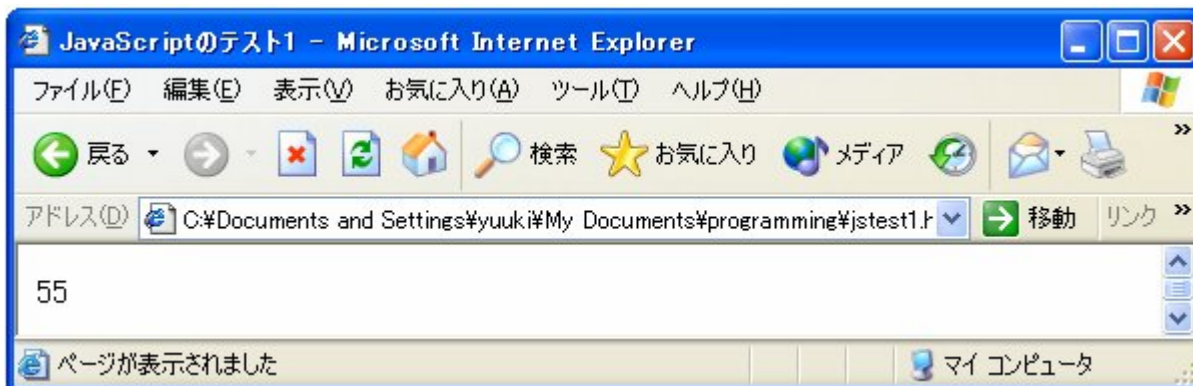
```
for (初期処理式; 条件式; 再処理式){  
  ステートメント1;  
  ステートメント2;  
  .....  
}
```

2-4-2 for文(2)

- for文の簡単な使用例

```
var sum=0;
var i;
for (i=1; i<=10; i++){
  sum+=i;
}
document.write(sum,"<br>");
```

実行結果



2-4-2 for文(3)

- for文を用いて二重ループを作成したプログラム(スクリプト部分)

```
document.write("<center>");
document.write("<h1>九九の表</h1>");
document.write("<table border>");
for (var i=1;i<10;i++){
  document.write("<tr>");
  //ひとつの行を作成
  for(var j=1;j<10;j++){
    document.write("<td>");
    if( i==1 || j==1 ){
      document.write("<b>",i*j,"</b>");
    }else{
      //掛け算の結果を表示
      document.write(i*j);
    }
    document.write("</td>");
  }
  document.write("</tr>");
}
document.write("</table>");
document.write("</center>");
```

プログラムの流れをfor文やif文などの制御構造と簡単な言葉で記述することを「疑似コーディング」と呼ぶ

2-4-2 for文(4)

- 実行結果



The screenshot shows a Microsoft Internet Explorer browser window titled "JavaScriptのテスト1 - Microsoft Internet Explorer". The address bar shows the file path "C:\Documents and Settings\yuuki\My Documents\programming\jstest1.t...". The main content area displays a 9x9 multiplication table with the title "九九の表" centered above it. The table contains the following data:

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

The status bar at the bottom of the browser window shows "ページが表示されました" (Page displayed) and "マイ コンピューター" (My Computer).

2-4-3 for~in文 (1)

- for~in文 (for文の特殊型)

書式

```
for (変数名 in オブジェクト){  
    ステートメント  
    .....  
}
```

オブジェクトのプロパティを順に変数名に格納して「{}」で囲まれたブロック内のステートメントを実行する

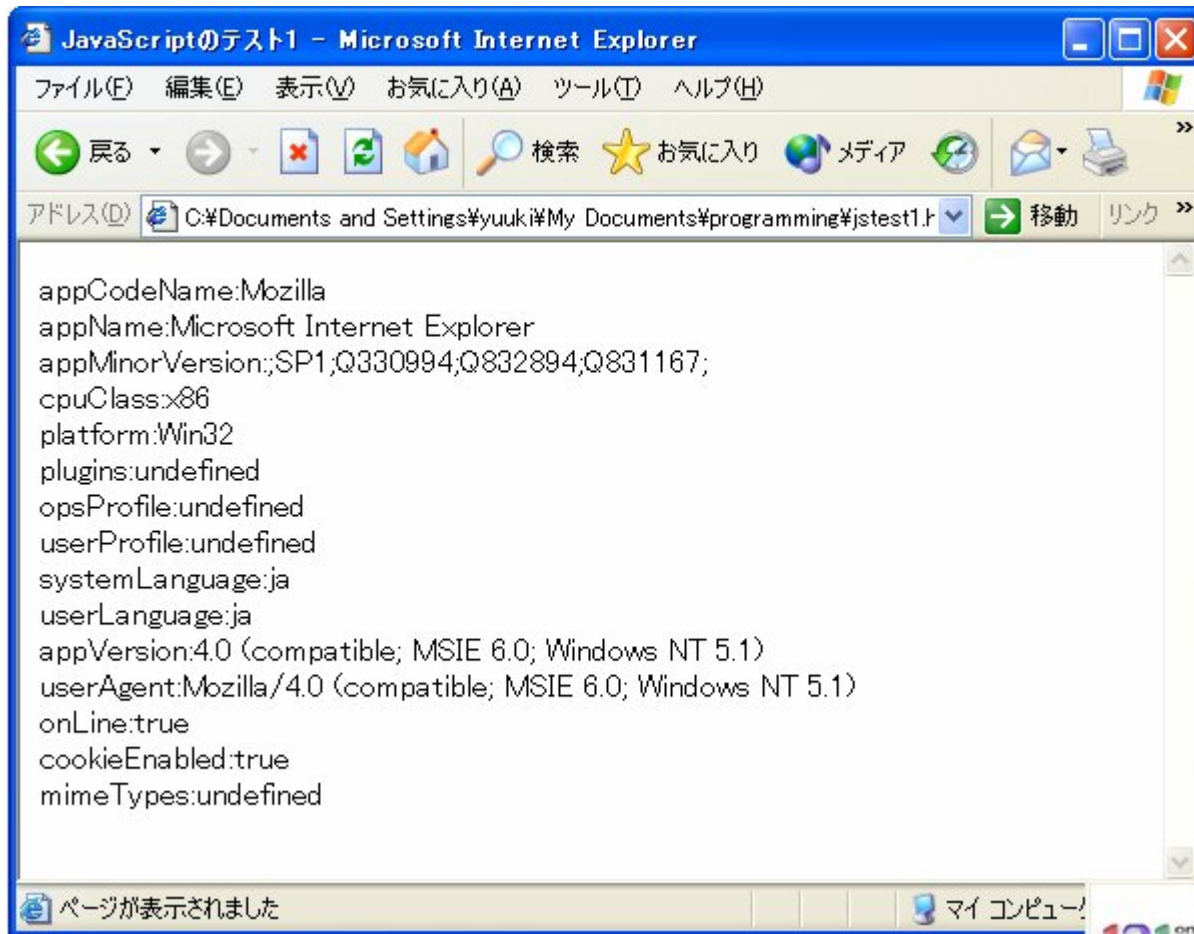
- for~in文を使用した例 (スクリプト部分)

```
var prop;  
for (prop in navigator){  
    document.write(prop,":",navigator[prop],"<br>");  
}
```

navigatorオブジェクトはWebブラウザに関するプロパティを保持

2-4-3 for~in文 (2)

- 実行結果



The screenshot shows a Microsoft Internet Explorer window titled "JavaScriptのテスト1 - Microsoft Internet Explorer". The address bar displays the file path "C:\Documents and Settings\yuuki\My Documents\programming\jstest1.js". The main content area displays the following JavaScript test results:

```
appCodeName:Mozilla
appName:Microsoft Internet Explorer
appMinorVersion:;SP1;Q330994;Q832894;Q831167;
cpuClass:x86
platform:Win32
plugins:undefined
opsProfile:undefined
userProfile:undefined
systemLanguage:ja
userLanguage:ja
appVersion:4.0 (compatible; MSIE 6.0; Windows NT 5.1)
userAgent:Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
onLine:true
cookieEnabled:true
mimeTypes:undefined
```

The status bar at the bottom indicates "ページが表示されました" (Page displayed) and "マイ コンピュータ" (My Computer).

2-4-4 while文(1)

- while文

ある条件が真である間、ループ内の処理を実行し続ける制御構造

書式

```
while (条件式) ステートメント;
```

あるいは

書式

```
while (条件式){  
ステートメント1;  
ステートメント2;  
.....  
}
```

2-4-2 while文(2)

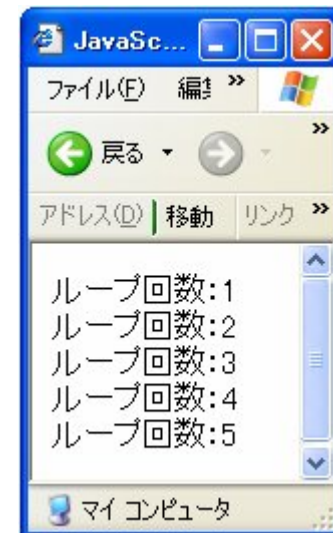
- 1から10までの総和を求めるプログラム(スクリプト部分)

```
var sum=0;
var i=1;
while(i <= 10){
sum+=i;
i++;
}
```

- ループの中断break文

break文を用いたプログラム(スクリプト部分)

```
//iの値を1ずつ増やし5になったらループを抜ける
var i=0;
while(true){
i++;
document.write("ループ回数:",i,"<br>");
if(i==5)break;
}
```

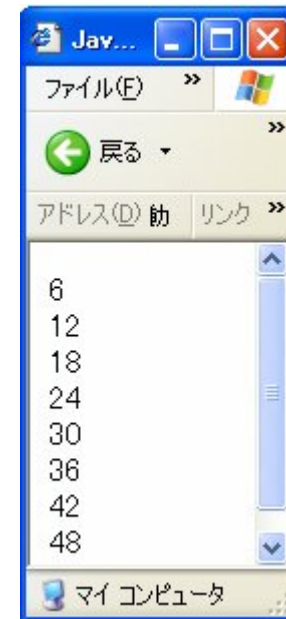


2-4-2 while文(3)

- ループの中断continue文

ループの途中で作業を中断して、強制的に次の繰り返しを開始させる。

```
//6で割り切れる数を表示  
var max=100;  
for(var i=1;i<max;i++){  
  if((i%6)!=0)continue;  
  document.write(i,"<br>");  
}
```



2-4-5 try~catch文 (1)

- try~catch文

処理をトライ (try) して、もし例外が発生したらキャッチ (catch) する
例外とは、プログラムの実行中に発生する何らかのエラーのこと

書式

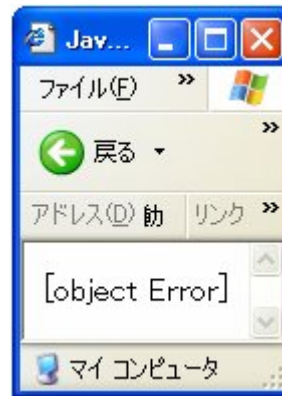
```
try{  
  ステートメント;  
  .....  
}catch(e){  
  ステートメント;  
  .....  
}
```

例外の内容はcatch文の()内に記述した変数(この場合e)に格納される。

2-4-5 try~catch文 (2)

- try~catch文の例

```
try{
var today=new Date();
today.setmonth(3);
}catch(e){
document.write(e);
}
```



月を設定するために「setMonth()」メソッドを呼び出す
ここではsetmonth()と記述ミスをしているため例外が発生

練習問題

1.for文を用いて下の図のように表示するプログラムを作成せよ。



練習問題2

- while文を用いて下図のような数字あてゲームを作成せよ。

