

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

ソフトウェア開発 第 7 回目授業

平野 照比古

2017/11/10

C 言語の変数名にマッチする

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ C 言語の変数名 (正確には識別子) は英字で始まり、そのあとに英数字が並んだもの (正確にはもう少し使える文字がある)
- ▶ 先頭の文字は文字クラスを使うと [A-Za-z]
- ▶ そのあとは英数字
- ▶ その文字クラスは `\w`
- ▶ 0 個でもよいので、`\w*`
- ▶ 全体がこれだけであることを保証するためには位置指定子をつける

^[A-Za-z]\w*\$

浮動小数リテラルをにマッチする正規表現 (1)

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

浮動小数リテラルは次の部分から成り立っている。

[符号][整数部][小数点][小数部][指数部]

このうち、[符号] や小数点以下の部分はなくてもよい。

浮動小数リテラルをにマッチする正規表現 (2)

- ▶ 符号部は+または-からなる一文字からなる。一度だけまで現れてよいので、この部分は `[+-]?` または `(+|-)?` で表される。
- ▶ 整数部は 10 進数の並びであり最低 1 文字は必要であるので反復の指定は+となる。したがって、この部分は `\d+` で表される。
- ▶ 小数点 `.` は正規表現では任意の文字にマッチするのでエスケープする必要がある。したがってこの部分は `[\.]` となる。
- ▶ 小数部は数字が並べられる。全くなくてもよいので反復の指定は*となる。

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

浮動小数リテラルをにマッチする正規表現 (3)

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ 指数部は指数の開始を表す文字 E または e で始まる 10 進数である。数字は最低一つ必要であるのでこの部分は $(E|e)\backslash d^+$ となる。
- ▶ これらを合わせると求める正規表現が得られる。小数部などがなくてもよいのでそれらの部分には反復指定 $?$ を付ければよい。

$^{\wedge} [+-]? \backslash d^+ (\backslash . \backslash d^*)? ((E|e) [+-]? \backslash d^+)? \$$

浮動小数リテラルをにマッチする正規表現 (4)

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

正式な数値リテラルでは小数点の前に整数部がない .1 など
も許しているが、ここではマッチしない。

24 時間制の時刻の表し方

時、分、秒はすべて 2 桁とし、それらの区切りは:

- ▶ 時間は 00 から 23 までであるので時間の初めの文字が 0 と 1 のときと、2 のときで分ける必要がある。
- ▶ 時間の先頭が 0 と 1 のときはそのあとの文字は 0 から 9 まで取れるので、`[01]\d` となる。
- ▶ 2 ではじまるときは 0 から 3 まで取れるので、`2[0-3]` となる。
- ▶ 同様に、分と秒は先頭の文字が 0 から 5 までであるので `[0-5]\d` となる。
- ▶ | の範囲を限定するため時間のところの () を忘れないこと。

求めるものは次のとおりである。

`^([01]\d|2[0-3]):[0-5]\d:[0-5]\d$`

`^([01]\d|2[0-3])(:[0-5]\d){2}$`

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

日付を分ける-ほしい部分を指定 (match を利用)

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ `/(\d+) 年 (\d+) 月 (\d+) 日/`
 - ▶ 戻り値の 1, 2, 3 番目に求める値が入る
 - ▶ したがって分割代入 `[,y,m,d]` で変数に代入ができる。
- ▶ この形ならば `/\d+/g` で十分

日付を分ける-区切る部分を指定 (split を利用)

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ /年|月|日/
戻り値の 0, 1, 2 番目に求める値が入る
- ▶ 文字クラスでもよいが「平成」が付いた場合には対応できない
- ▶ /\D+/(数字以外で分ける)

```
"aaaabaaabb".match(/.*b/);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10
a	a	a	a	b	a	a	a	b	b

```
["aaaabaaabb"]
```

- ▶ 最後に b が来る文字列がマッチ
- ▶ .*は貪欲であるのでできるだけ長い任意の文字列とマッチする
- ▶ 最後の b は 10 番目のものが対応

```
"aaaabaaabb".match(/.*b/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

```
["aaaabaaabb"]
```

前と同じ理由により、結果は同じ

```
"aaaabaaabb".match(/.*?b/);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10
a	a	a	a	b	a	a	a	b	b

```
["aaaab"]
```

- ▶ bで終わる文字列を表している
- ▶ その前の.*?は非貪欲なマッチをする
- ▶ 最後のbは5番目のもの

```
"aaaabaaabb".match(/.*?b/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10
a	a	a	a	b	a	a	a	b	b

```
["aaaab", "aaab", "b"]
```

- ▶ 前と同様に非貪欲なマッチング
- ▶ g オプションがついているので一つ目以降見つけた位置から再度マッチするものを探す
- ▶ 全体で 3 つ答えを返す

```
"abccbckkccaaMMaacc".match(/((.)\2).*\1/);
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
a b c c b c c k k c c a a M M a a c c

```
["ccbckkccaaMMaacc", "cc", "c"]
```

- ▶ 正規表現の (.) は左から数えて 2 番目の括弧になる
- ▶ この部分にマッチした文字は \2 で参照できる
- ▶ (.)\2 は同じ文字が 2 つ並ぶものにマッチ
- ▶ この部分全体が再び括弧でくくられているので、この部分が \1 で参照できる
- ▶ この正規表現は同じ 2 つの文字で挟まれた文字列にマッチする
- ▶ 中央部の正規表現は任意の文字列を表す
- ▶ 貪欲なマッチなのではじめに現れる同じ文字が 2 つ続く 3,4 番目の cc が一番最後に現れる 18 番目と 19 番目の cc と組み合わせられる
- ▶ グローバルな検索ではないときには戻り値に \1 と \2 が含まれる。

```
"abccbckkccaaMMaacc".match(/((.)\2).*\1/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	a	a	c	c

```
["ccbckkccaaMMaacc"]
```

- ▶ g フラグが付いているが、条件に合うものは一つしかない
- ▶ 戻り値に \1 と \2 が含まれない

```
"abccbckkccaaMMaacc".match(/((.)\2).*?\1/);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	a	a	c	c

```
["ccbcc", "cc", "c"]
```

- ▶ 前問と異なり、中央部の任意の文字列が非貪欲になっている
- ▶ 3, 4 番目と 6, 7 番目の cc が対応
- ▶ 戻り値の配列の 2 番目と 3 番目は \1 と \2


```
"abccbcckkccaaMMaacc".match(/((.)\2).*?\1/g);
```

ソフトウェア開発
第 7 回授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	a	a	c	c

```
["ccbcc", "ccaaMMaacc"]
```

g が付いているのでさらに 10 番目と 11 番目、18 番目と 19 番目の cc が対応

```
"abccbckkccaaMMaa".match(/((.)\2).*\1/);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	a	a

```
["ccbckkcc", "cc", "c"]
```

- ▶ これまでの文字列から最後の 2 文字を取り除いた文字列で同じことを行っている
- ▶ 3,4 番目の cc に対応するのは 11, 12 番目のもの

。

```
"abccbckkccaaMMaa".match(/((.)\2).*\1/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	a	a

```
["ccbckkcc", "aaMMaa"]
```

- ▶ g フラグが付いているのでマッチした部分列が返される
- ▶ 3,4 番目のと 10,11 番目の cc が対応
- ▶ そのあとの部分列で 12,13 番目と 16,17 番目の aa が対応する
- ▶ マッチした部分列は 2 つ

```
"abccbckkccaaMMccaa".match(/((.)\2).*\1/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	c	c	a	a

```
["ccbckkccaaMMcc"]
```

- ▶ これまでの文字列と最後の 4 つが入れ替わっている
- ▶ 3,4 番目と 16,17 番目の cc が対応

```
"abccbckkccaaMMccaa".match(/((.)\2).*?\1/g);
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	c	b	c	c	k	k	c	c	a	a	M	M	c	c	a	a

```
["ccbcc", "ccaaMMcc"]
```

- ▶ 前問と異なり、非貪欲な任意の部分文字列を途中に取る
- ▶ 3, 4 番目と 6, 7 番目の cc、10,11 番目と 16,17 番目の cc が対応する

正規表現を作る

目付を得る

正規表現のマッチを確認する

正規表現の利用

DOM とは

DOM のメソッド

DOM のプロパティ

```
function Person(name, y, m, d){
    if(name === "") throw new Error("名前がありません");
    this.name = name;
    this.year = y;
    if(m<1 || m>12) throw new Error("月が不正です");
    var date = new Date(y,m,0);
    if(d<1 || d>date.getDate()) throw new Error("日が不正です");
    this.month = m,
    this.day = d
}
Person.prototype = {
...
}
```

このリストでは十分なエラー処理がなされていない。

- ▶ 数に変換されない文字列の入力に対しては値が NaN になる。
- ▶ その結果、たとえば月の値の評価 $m < 1$ は false となり、エラーチェックを通り抜けてしまう。
- ▶ 対処法としては $!(m \geq 1 \ \&\& \ m \leq 12)$ とすることも考えられるが、それでも小数点付きの数が排除できない。
- ▶ Web アプリケーションではテキストボックスの入力は文字列になるので、文字列の段階でチェックするほうが楽

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

Person オブジェクトの範囲を外部から入力させるときに文字列を数に変換する前に数値リテラルになっているかを判定することでプログラムが不正な値を受け付けないようにできる。

整数値だけにするのであれば正規表現は/[^]\d+\$/である。

Person2 の修正

```
<title>エラーオブジェクト (改良)</title>
function Person(name, y, m, d){
    if(name === "") throw new Error("名前がありません");
    this.name = name;
    this.year = checkNum(y,1900,2020,"年");
    this.month = checkNum(m,1,12,"月");
    var date = new Date(y,this.month,0);
    this.day = checkNum(d,1,date.getDate(),"日");
}
function checkNum(s, low, high, mes) {
    if(s.match(/^\d+$/)) {
        if(s>=low && s<=high) return s-0;//文字列を数字に変換
    }
    throw new Error(mes+"が不正です");
}

...
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

`prompt()` で戻ってきた文字列が数字だけからなっているかをチェックしたうえで、与えられた範囲内にあるかを調べている。

外部入力のチェックを！

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ 外部からのデータの入力に対しては、データを吟味してから利用する
- ▶ 特に Web ページのテキストボックスからのデータ入力を利用して不正行為を行う手法が知られている

次のリストは Google Maps を利用して地図を表示するものである。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>初めての GoogleMaps</title>
6 <script type="text/javascript"
7     src="http://maps.google.com/maps/api/js?sensor=false"></script>
```

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

平野 照比古

```

8 <script type="text/javascript">
9   window.onload = function() {
10     let latlng = new google.maps.LatLng(35.486210,139.341443);
11     let myOptions = {
12       zoom: 10,
13       center: latlng,
14       mapTypeId: google.maps.MapTypeId.ROADMAP
15     };
16     let mapCanvas = document.getElementById("map_canvas")
17     let map = new google.maps.Map(mapCanvas, myOptions);
18   }
19 </script>
20 <link rel="stylesheet" type="text/css" href="map.css" />
21 </head>
22 <body>
23   <div id="map_canvas" ></div>
24 </body>
25 </html>

```

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

DOM とは

DOM のメソッド

DOM のプロパティ

リストの解説 (1)

ソフトウェア開発
第 7 回目授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ 1 行目は HTML 文書の DOCTYPE 宣言である。この形は HTML5 におけるもの
- ▶ 2 行目はこの HTML 文書のルート要素と呼ばれるものである。最後の 26 行目の `</html>` まだが有効となる。
- ▶ すべての要素はこの範囲になければならない。

リストの解説 (2) - <head>

3 行目から始まる<head>はブラウザに表示されない、いろいろな HTML 文書の情報を表す。

- ▶ 4,5 行目はこの文書の形式や文字集合を記述している。ここでは内容は text/html の形式、つまり、テキストで書かれた html の形式で書かれていることを表す。¹
- ▶ 6 行目の<title>はブラウザのタブに表示される文字列を指定している。
- ▶ 6,7 行目は Google Maps のライブラリーを読み込むためのものである。このように JavaScript のプログラムは外部ファイルとすることができる。
- ▶ 9 行目から 19 行目は HTML 文書内に書かれた JavaScript である。詳しい解説は後の授業で行う。
- ▶ 20 行目は HTML 文書の見栄えなどを規定する CSS ファイルを外部から読み込むことをしている。

¹このような方法でファイルのデータ形式を表すことを MIME(Multipurpose Internet Mail Extension) タイプと呼ぶ。元来、テキストデータしか扱えない電子メールに様々なフォーマットのデータを扱えるようにする規格である。

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

HTML 文書で実際にブラウザ内で表示される情報は
<body>要素内に現れる。

- ▶ このリストでは Google Maps を表示するための
<div>要素が一つあるだけである。このとき、<div>は
<body>の子要素であるといい、<body>は<div>の親要素という。
- ▶ 各要素名または要素の終了を示すタグ (<...>) の間に
文字列がある場合、その部分はテキストノードと呼ば
れるノードが作成されている。

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

各要素は<との中に現れる。初めに現れる文字列が要素名であり、そのあとに属性と属性値の組がいくつか並ぶ。

- ▶ 属性とその属性値は=で結ばれる。
- ▶ 属性値は”ではさまれた文字列として記述
- ▶ <script>要素では属性 type と src が設定
- ▶ 24 行目の<div>要素では属性 id に属性値 map_canvas を設定している。なお、この要素は CSS によっても属性が定義されている。

20 行目で参照している CSS ファイル

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

```
1 #map_canvas{
2     width:500px;
3     height:500px;
4     float:left;
5     margin:5px 10px 5px 10px;
6 }
```

- ▶ CSS の各構成要素は HTML 文書の要素を選択するセレクタ (ここでは#map_canvas) とそれに対する属性値の並び ([属性]:[属性値];) からなる。
- ▶ #で始まるセレクタはそのあとの文字列を<id>の属性値に持つ要素に適用される。
- ▶ したがって、ここの規則は 24 行目の<div>要素に適用
- ▶ その内容は Google Maps が表示される画面の大きさ (width と height)、配置の位置 (float) と要素の外に配置される空白 (margin) を指定

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

- ▶ カスケーディングスタイルシート (CSS) は HTML 文書の要素の表示方法を指定するもの
- ▶ CSS は JavaScript から制御可能
- ▶ 文書のある要素に適用されるスタイルルールは、複数の異なるルールを結合 (カスケード) したもの
- ▶ スタイルを適用するためには要素を選択するセクタで選ぶ。

セレクトの種類は配布資料を参照のこといくつか注意する点を挙げる。

- ▶ 属性 `id` の属性値の前に `#` をつけることでその要素が選ばれる。
- ▶ 属性 `class` の属性値の前に `.` をつけることでその要素が選ばれる。
- ▶ `nth-child(n)` には単純な式を書くことができる。このセレクトは複数書いてもよい。
- ▶ `E F` と `E > F` の違いを理解しておくこと。たとえば `div div` というセレクトは途中で別の要素が挟まれていてもよい。また、`<div>` 要素が 3 つある場合にはどのような 2 つの組み合わせも対象となる。

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

- ▶ Document Object Model(DOM) は HTML 文書などの要素をノードとしたツリー構造で管理する方法
- ▶ DOM のメソッドやプロパティを使うことで各要素にアクセスしたり、属性値やツリーの構造を変化させることが可能
- ▶ DOM の構造は開発者ツールなどで見ることができる。

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ Google Chrome では開発者ツールから Elements タブで確認する。ここで要素上で右クリックして Edit as HTML を選択するとテキストとして編集できる。
- ▶ FireFox では開発ツールから「開発ツールを表示」を選択し、インスペクタタブで DOM ツリーが確認できる。要素上で右クリックから「HTML として編集」とするとテキストとして編集できる。
- ▶ IE では開発者ツールを開き左にあるタブの一番上にある DOM Explorer で同様のことができる。

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

DOM では DOM ツリーを操作するためにメソッドやプロパティが規定されている。メソッドとはそのオブジェクトに対する操作である。次のような手段を提供している。

- ▶ 条件に合う要素または要素のリストを得る。
- ▶ 要素の属性を参照、変更ができる。
- ▶ 要素を新規に作成する。
- ▶ ある要素に子要素を追加したり、取り除いたりする。

条件に合う要素または要素のリストを得る (1)

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ `getElementById(id)`
属性 `id` の値が引数 `id` である要素を得る。
- ▶ `getElementsByTagName(Name)`
要素名が `Name` である要素のリストを得る。リストの各要素は配列と同様に [] で参照できる。
- ▶ `getElementsByClassName(Name)`
属性 `class` の値が `Name` である要素のリストを得る。リストの各要素は配列と同様に [] で参照できる。
- ▶ `getElementsByName(Name)`
属性 `name` が `Name` である要素のリストを得る。得られた各要素は配列と同様に [] で参照できる。

条件に合う要素または要素のリストを得る (2)

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ `querySelector(selectors)`
`selectors` で指定された CSS のセレクトタに該当する一番初めの要素を得る
- ▶ `querySelectorAll(selectors)`
`selectors` で指定された CSS のセレクトタに該当する要素のリストを得る。配列と同様に [] で参照できる

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ `getAttribute(Attrib)`
対象要素の属性 `Attrib` の値を得る。戻り値の型は文字列である。
- ▶ `setAttribute(Attrib,Val)`
対象要素の属性 `Attrib` の値を `Val` にする。数を渡しても文字列に変換される。
- ▶ `hasAttribute(Attrib)`
対象要素に属性 `Attrib` がある場合は `true` を、ない場合は `false` を返す。
- ▶ `removeAttribute(Attrib)`
対象要素の属性 `Attrib` を削除

要素を新規に作成する

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ `createElement(Name)`
Name で指定した要素を作成
- ▶ `createElementNS(NS,Name)`
名前空間 NS で定義されている要素 Name を作成
- ▶ `createTextNode(text)`
text を持つテキストノードを作成
- ▶ `cloneNode(bool)`
bool が true のときは対象要素の子要素すべてを、
false のときは対象要素だけの複製を作る。

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

- ▶ 指定した要素が定義されている規格を指定するもの
- ▶ 一つの文書内で複数の規格を使用する場合、作成する要素がどこで定義されているのかを指定
- ▶ 異なる規格で同じ要素名が定義されていてもそれらを区別することが可能
- ▶ 通常の HTML 文書では
`http://www.w3.org/1999/xhtml` を指定

プルダウンメニューの例

1月から12月までを選択できるプルダウンメニューを作成

```
<body>
  <form id="menu">
    <select>
      <option value="1">1 月</option>
      <option value="2">2 月</option>
      <option value="3">3 月</option>
      <option value="4">4 月</option>
      <option value="5">5 月</option>
      <option value="6">6 月</option>
      <option value="7">7 月</option>
      <option value="8">8 月</option>
      <option value="9">9 月</option>
      <option value="10">10 月</option>
      <option value="11">11 月</option>
      <option value="12">12 月</option>
    </select>
  </form>
</body>
```

正規表現を作る

目付を得る

正規表現のマッチを確認する

正規表現の利用

DOM とは

DOM のメソッド

DOM のプロパティ

- ▶ ユーザからの入力を受け付ける要素は通常、`<form>`要素内に記述
- ▶ プルダウンメニュー の要素名は`<select>`
- ▶ 選択する内容は`<option>`要素
- ▶ `<option>`要素の属性 `value` の値が選択した値として利用できる。
- ▶ `<option>`要素内の文字列 (テキストノード) がプルダウンメニューに表示される
- ▶ `<select>`は`<form>`の子要素であり、各`<option>`は`<select>`の子要素

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

プルダウンメニュー——プログラムで作成 (1)

ソフトウェア開発
第 7 回授業

平野 照比古

前回の演習

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

```
7 //<![CDATA[
8   window.onload = function(){
9       let Select = document.createElement("select");
10      document.getElementById("menu").appendChild(Select);
11      for(let i=1;i<=12;i++) {
```


プルダウンメニュー——プログラムで作成 (2)

ソフトウェア開発
第 7 回目授業

平野 照比古

```
12     let Option = document.createElement("option");
13     Option.setAttribute("value",i);
14     Select.appendChild(Option);
15     Option.appendChild(document.createTextNode(i+"月"));
16 }
17 }
18 //]]>
19 </script>
20 </head>
21 <body>
22     <form id="menu"></form>
23 </body>
24 </html>
```

前回の演習

正規表現を作る
日付を得る
正規表現のマッチを確認する
正規表現の利用

HTML 文書の構成

DOM の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

Table: DOM 要素に対するプロパティ(1)

プロパティ名	説明
firstChild	指定された要素の先頭にある子要素
lastChild	指定された要素の最後にある子要素
nextSibling	指定された子要素の次の要素
previousSibling	現在の子要素の前にある要素
parentNode	現在の要素の親要素
hasChildNodes	その要素が子要素を持つかどうか
nodeName	その要素の要素名前
nodeType	要素の種類 (1 は普通の要素、3 はテキストノード)
nodeValue	(テキスト) ノードの値
childNodes	子要素の配列

正規表現を作る

日付を得る

正規表現のマッチを確認する

正規表現の利用

DOM とは

DOM のメソッド

DOM のプロパティ

Table: DOM 要素に対するプロパティ(2)—DOM4

プロパティ名	説明
children	子要素のうち通常の子要素だけからなる要素の配列
firstElementChild	指定された要素の先頭にある通常の子要素
lastElementChild	指定された要素の最後にある通常の子要素
nextElementSibling	指定された子要素の次の通常の子要素
previousElementSibling	現在の子要素の前にある通常の子要素