

[目次へ](#)

## WEB編 HTMLのリンクのつくりかた

---

[TODO: 動物さん]

ここで今さら説明するのも何ですが、HTMLどうしのリンクを作るやり方を確認しておきます。つまりAタグの書き方を説明するだけです。分かっている方は飛ばしてもいいですよ。

WEBで色々なサイトを見ていて、下線の付いた文字をクリックすると別のどこかに飛べますね。あれのことです。あれを表現するには、HTMLでは下のように入ります。

```
<a href="http://www.yahoo.co.jp/">ここをクリック</a>
```

上のような書き方をしておくと、「ここをクリック」という文字に下線がついて、マウスをクリックすると、href=" " の中に書いてあるアドレスにジャンプすることができます。http:... とかで始まる好きなURLを中に書けばいいです。

今のを、絶対URL指定っていいいます。これの他に、相対URL指定ってのがあります。下みたいな感じ。

```
<a href="tonari.html">となりのページへ</a>
```

これをクリックすると、今表示しているHTMLページと同じフォルダの中にある tonari.html というページに移動することができます。もちろん、あらかじめ作っておかなきゃだめですけど。

内輪のページ内の移動なら相対URLで、どこか別のサイトに移動する必要があるら絶対URL、ってというのがよくある使い方ですね。

同じフォルダにあるページじゃなくて、ひとつ上のフォルダにさかのぼってリンクを作る方法とかもあります。さしあたって必要でないので、まだ説明しません。

で、この相対URLは、CGIスクリプトの画面遷移を表す時も使えます。

今表示しているのが、たとえば `http://localhost/cgi-bin/page.cgi?number=3` [`http://localhost/cgi-bin/page.cgi?number=3`] という感じだとしましょう。きっと、何かの3ページ目でも表示しているのかなと推測できそうですね。ここから「前ページへ」「次ページへ」なんていうリンクを提供して、そこをマウスクリックするとそこに移動（遷移）できるようにするにはどうするでしょう。

下のようなリンクを作ればいいです。

```
<a href="page.cgi?number=2">前ページへ</a>
```

```
<a href="page.cgi?number=4">次ページへ</a>
```

リンクの中身に、「?」以降のパラメータ部分まで書き込んでしまえばいいというわけです。こうすると、リンクをクリックするたびに、同じCGIスクリプトなんだけど、そのつど色々な条件で動作するという感じのものが作れることになりますね。CGIスクリプトの中でリンクのためのAタグをうまくこしらえてprintする必要が出てきますが、ここらへんは普通の文字列操作です。それほど困ることはないでしょう。

ここまでよろしいでしょうか。これを踏まえて、ちょっとホネのある練習問題をやってみようかと思えます。なんと、カレンダーシステムを作ります。ちゃんと表形式で表示され、先月と来月に進むための機能があって、しかも日曜日は赤く表示されるという美しい設計。さあやるぞ。

とその前に、カレンダーを表示するためのちょっと便利なモジュールの助けを得ることを考えましょう。その名も calendar モジュールというのがあって、こいつを使うと楽できますよ。っていうか、こいつなしではやる気も起こらない。

## calendarモジュール

---

calendarモジュールは、datetimeモジュールの親戚みたいなものですが、特にカレンダー的にまとまった情報を扱うときに強力なものです。

ちょっとの間、対話シェルからいろいろと試して遊びましょう。

ところでここはあまり重要でもないですが、新しくモジュールを扱い始めるときには、dir命令を使って、モジュールで何ができるようになりそうかを眺めてみるというのは悪くない習慣だと思います。下のように入力してみてください。

```
>>> import calendar
>>> dir(calendar)
```

この結果ズラズラと表示されるものが、calendarモジュールの中身として使える何かです。この中には関数もあれば変数（定数）もあれば、またはモジュール内で入れ子式にインポートしているモジュールなんかもあって、すべてが何が具体的な機能を持っているとは限りません。でも、「ああ、calendarモジュールにはsetfirstweekdayなんていう関数があったっけ」とか思い出すための材料としては有用なものなので、こんなやりかたを覚えておいてください。

さて、手始めに、2010年2月のカレンダーをいきなり出力してみしましょう。下のような命令を与えるだけです。

```
>>> calendar.month(2010, 2)
```

この結果はちょっとダメですね。改行文字が '\n' といった様子で直接出力されます。print を使って表示しなおしてください。

```
>>> print calendar.month(2010, 2)
February 2010
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

ちょっと感動的ではないでしょうか。完全なカレンダーです。

とはいえ、文字列としてこれを得られても、こいつを改めてHTMLに整形することを考えると、そんなに便利なものではありません。プログラ的には、calendar.month ではなくて、calendar.monthcalendar を使うのがいいと思います。下の例。

```
>>> calendar.monthcalendar(2010, 2)
[[1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12, 13, 14], [15, 16, 17, 18, 19, 20, 21],
 [22, 23, 24, 25, 26, 27, 28]]
```

数字のリスト、のリストが得られます。見やすく直してみると、下のようリストですね。

```
[
 [ 1,  2,  3,  4,  5,  6,  7],
 [ 8,  9, 10, 11, 12, 13, 14],
 [15, 16, 17, 18, 19, 20, 21],
 [22, 23, 24, 25, 26, 27, 28]
]
```

なるほど、実際のカレンダーに対応しています。これならプログラムで扱うときにずっと都合がよいでしょう。

ところで、このカレンダーでは、左端が月曜日ですね。日本では左端を日曜日にするのが普通です。ここらへん、どう対応すればよいでしょうか。実は calendar モジュールはそれをあらかじめ指定できるようになっています。下のよう命令を実行してみましょう。

```
>>> calendar.setfirstweekday(calendar.SUNDAY)
>>> print calendar.month(2010, 2)
February 2010
Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6
  7  8  9 10 11 12 13
 14 15 16 17 18 19 20
 21 22 23 24 25 26 27
 28
```

このように、最初に setfirstweekday という命令を最初に一回実行しておけば、以後、週の最初は日曜日、という風に出力結果が変わってきます。リストで得られる monthcalendar も同様。

```
>>> print calendar.monthcalendar(2010, 2)
[
 [ 0,  1,  2,  3,  4,  5,  6],
 [ 7,  8,  9, 10, 11, 12, 13],
 [14, 15, 16, 17, 18, 19, 20],
 [21, 22, 23, 24, 25, 26, 27],
 [28,  0,  0,  0,  0,  0,  0]
]
(出力結果は適宜整形しています)
```

リストでカレンダーを得たときに、空白部分（前月や翌月にかぶる部分）は空白のかわりにゼロが入っているので注意してください。これはこの機能の仕様です。

ところで、calendar.SUNDAY って、単に 6 のことです。スクリプトで直接日曜日のことを 6 って書いても問題ないんですけど、calendar.SUNDAY と書いておいたほうが、あとで「ああ、週の初めは日曜日だね」と思いだせるので、タイピングの手間さえ惜しまないのなら、こっちの書き方のほうがいいと思います。まあ筆者もときどきサボって直接 6 とか書きますけどね。

さて、この「リストのリスト」をうまくHTMLに変換できるでしょうか。自力で考えてくれてもよいですが、例えば下のように書くといいですよ。この例ではカレンダーとは少し違って、3x3の表をつくるだけです。

```
cal = [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
print "<table>"
for wk in cal:
    print "<tr>"
    for dy in wk:
        if dy != 0:
            print "<td>%d</td>" % dy
        else:
            print "<td>--</td>"
    print "</tr>"
print "</table>"
```

ゼロのときは数字を出さない、という処理も書いてみました。また、スクリプト内の変数は wk が week、dy が day をそれぞれ表しています。week とか day とか書いても直接既存の関数にはかぶりませんが、日付を扱うモジュールを扱ってる時って、なんとなく意図しないところがかぶりそうな気がして、ついこうします。

あとは、Aタグを出力するときってどんな感じになるかな、というのと、「先月」「来月」ってのをどう導出しようかな、というあたりについても少しヒントを述べておきます。

2010年3月のカレンダーを表示するときのリンクは、たとえば自分自身のスクリプトが calendar.cgi だとしたら、calendar.cgi?year=2010&month=3 とかですね。こういうリンクをスクリプト内で作るときは、

```
link = '<a href="calendar.cgi?year=%d&month=%d">LINK HERE</a>' % (2010, 3)
print link
```

って感じですね。ちょっと込み入ってるように見えますが、注意して見れば大丈夫でしょう。

また、変数 y に年、変数 m に月がたとえば入っているとして、来月、先月はそれぞれどう計算できるかということ、

```
y = 2010
m = 3
next_y = y
next_m = m + 1
prev_y = y
prev_m = m - 1
```

とかいっただけの書き方では、12月の次が13月になってしまったりして不都合なことがわかりますね。現在が1月のときと、現在が12月のときには、特に注意して場合分けをしながら処理を行いましょう。

といったところで、大体ヒントはさしあげました。いよいよ、実際のCGIスクリプトを作ってみましょう。

スクリプトの中でやるべき処理は、

- 環境変数から、何年何月のカレンダーを出したいのかを検知する。この時点では、文字列としてデータが得られますので、intなどで実際の数字に変換する処理もきっと必要で

す。また、まともなパラメータが与えられなかった場合は今月とみなす、とかいった処理も必要かもしれません

- 前月へのリンクのためのURL文字列をつくる
- 来月へのリンクのためのURL文字列をつくる
- カレンダーをHTMLとして出力する
- それぞれのリンクもHTMLの一部として出力する

といったところですね。

うーん、できるでしょうか。できてくれると嬉しいんですが。

## 練習問題W02

---

【練習問題:W02】ひと月分のカレンダーを表示して、前月と翌月への遷移も可能なCGIスクリプトを書け。カレンダーを表示する際には、日曜日のみ文字色または背景色を変更せよ。