

Practice # 4

簡単なホームページをつくる

演習 4 A HTMLの基礎

2025.7.11

HTMLとは

狭義のHTMLとは

ブラウザ上で表示する文章の構造（各部の意味づけ）を与えるためのマークアップ言語

広義のHTMLとは

ブラウザ上で動作するアプリケーションを作成するための一連の言語技術の総称（HTML言語に対して各部の装飾を指示するCSS・各部（間）に動的な計算処理を指示するjavascriptを含めることがある）

（狭義の）HTML	<u>文章の意味付け</u>
CSS（スタイルシート）	<u>文章の装飾</u>
Javascript	<u>動的な計算処理</u>

狭義のHTML

...

```
<title>呼吸法を紹介するページ</title>
```

...

XXX.html

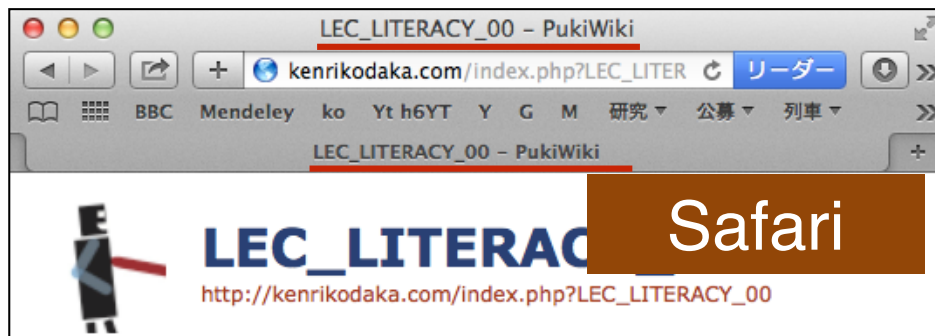
上の例は、文をtitleタグで囲むことによって、（呼吸法…）の部分が文章のタイトルであることを表している。

上記は、ドキュメント全体の構造においてタイトルとしての意味を与えているだけであり、ブラウザ上でどのように表示されるかについて、（狭義の）HTMLは無関心であることに注意。

実際の表示方法は、ブラウザの解釈に委ねられる。

実際に、titleの表示方法は、ブラウザによって異なる。

HTML5では、「タグを装飾的な機能から分離する」という思想がより明確とされている。



(狭義のHTML) はあくまで、文章の各部に意味を与えており、その表示方法はブラウザに委ねられる。

CSSはHTML文書を表示する際、どのような装飾を与えるかを、意味要素に対応させて細かく指定することができる。

```
...
<style>
  h1{color:red;}
</style>
...
<h1>見出しは赤色</h1>
...
```

指定可能な属性の例

サイズ

罫線

色

整列

フォント

* 上の例では、h1要素を持つ見出しの字の色を、赤色とするように指定しています。

統一性のあるデザインを持っているサイトは、CSSを活用していることが多い。

論文の基本的な情報

著者・タイトル・ジャーナル名 (引用形式)

Kodaka, K., & Ishihara, Y. (2014). Crossed hands strengthen and diversify proprioceptive drift in the self-touch illusion. *Frontiers in Human Neuroscience*, 8. doi:10.3389/fnhum.2014.00422

原文へのリンク

<http://journal.frontiersin.org/Journal/10.3389/fnhum.2014.00422/full>

Frontiersはオープンアクセス方式を採用している論文誌です。WEB上で誰もが自由に閲覧、あるいはPDFをダウンロードすることができます。

著者による論文の内容解説

著者による論文の解説 1 <身体が漂流する>

Javascript

HTMLと連携し、HTML文書の中に、複雑な計算処理・イベント処理（クリックやマウス動作に対する反応、タイマー処理など）を記述可能なスクリプト言語。

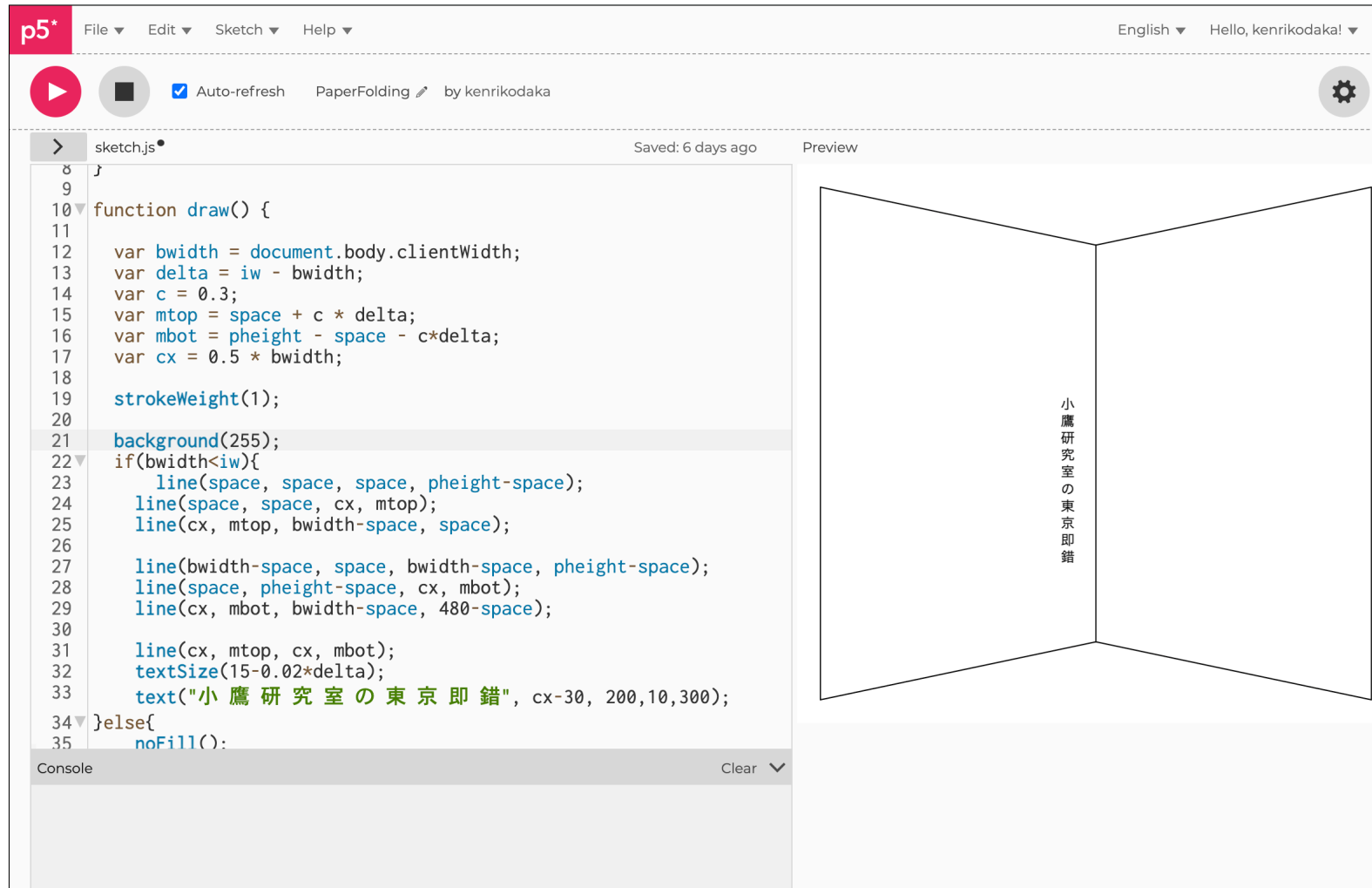


スクリプト言語は、コンパイルの必要がないため、記述と同時に即座に実行が可能。ただし処理速度の面では、通常のプログラム言語（C, JAVA⇨PROCESSINGなど）に劣る。

javascriptは、純正のHTMLコンポーネント。従って、ブラウザの動く環境であれば、どこでも実行可能。かつ、実行結果についても、OS間・ブラウザ間でブレが少ない。

Processing と HTML

Processingは、内部的にJavascriptに変換する仕組みを持っているため、作成したものをHTML5に埋め込むことが可能です。

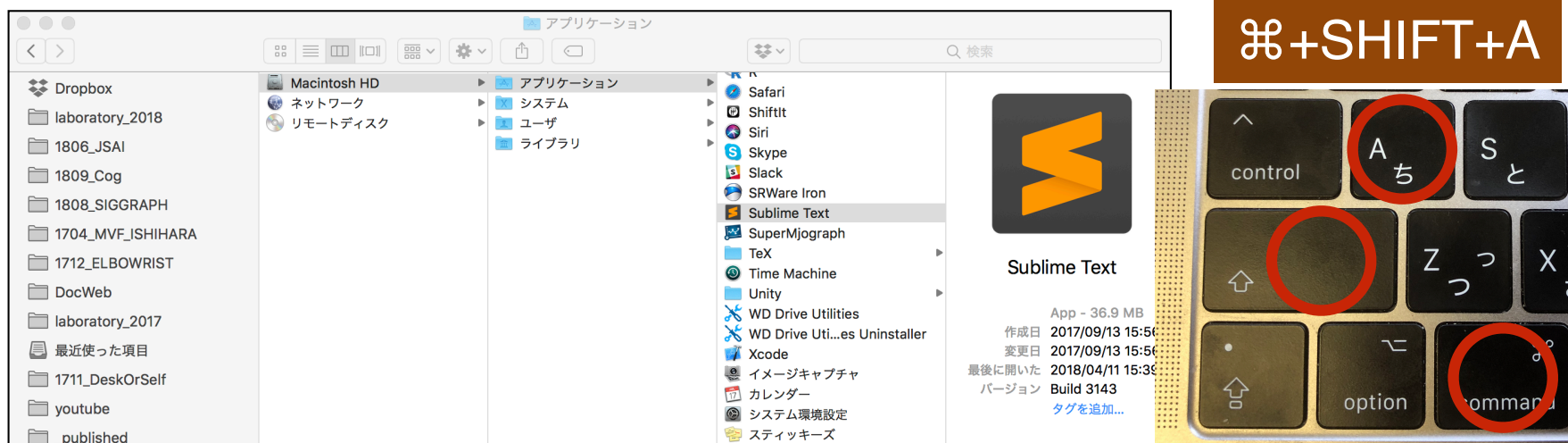


[p5js] <https://editor.p5js.org/>

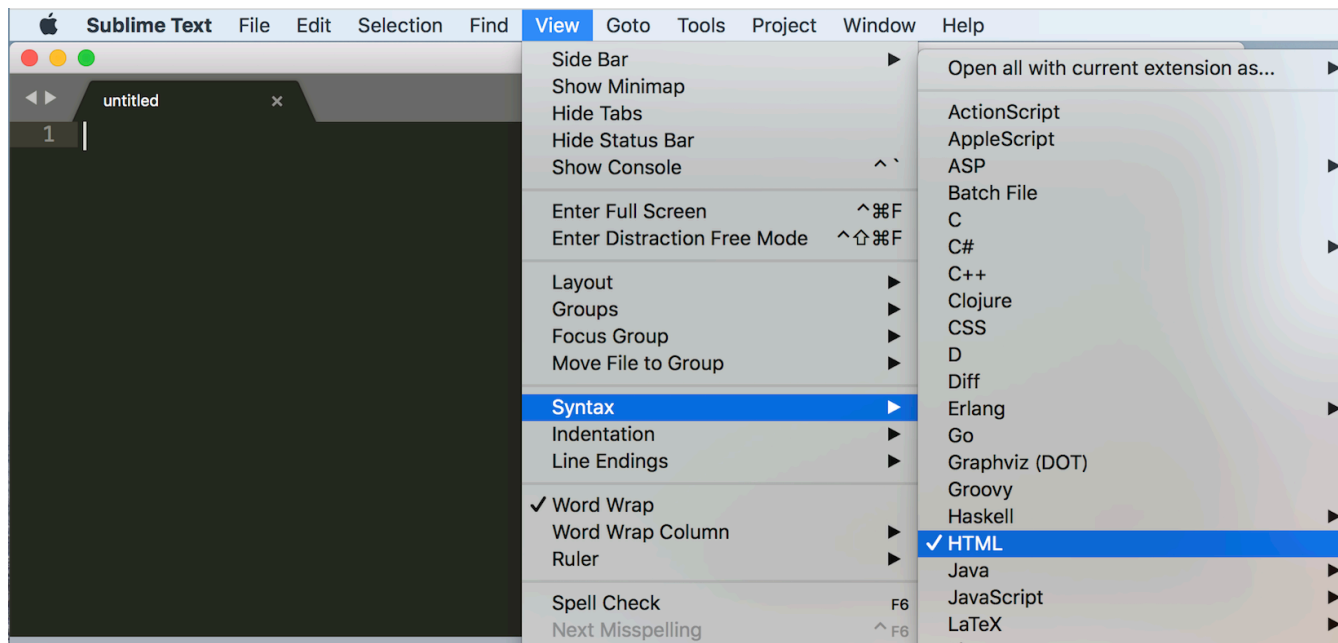
[Processing.js] <https://github.com/processing-js/processing-js>

エディタを開く

1) アプリケーションフォルダへ行って, Sublime Textを開く

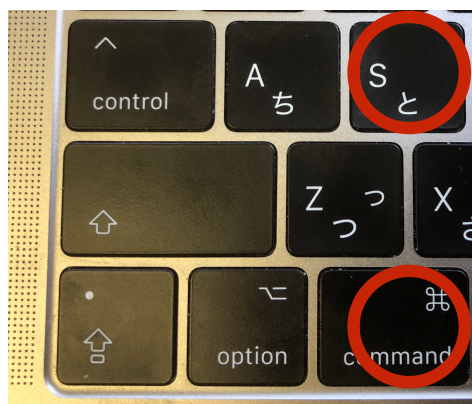
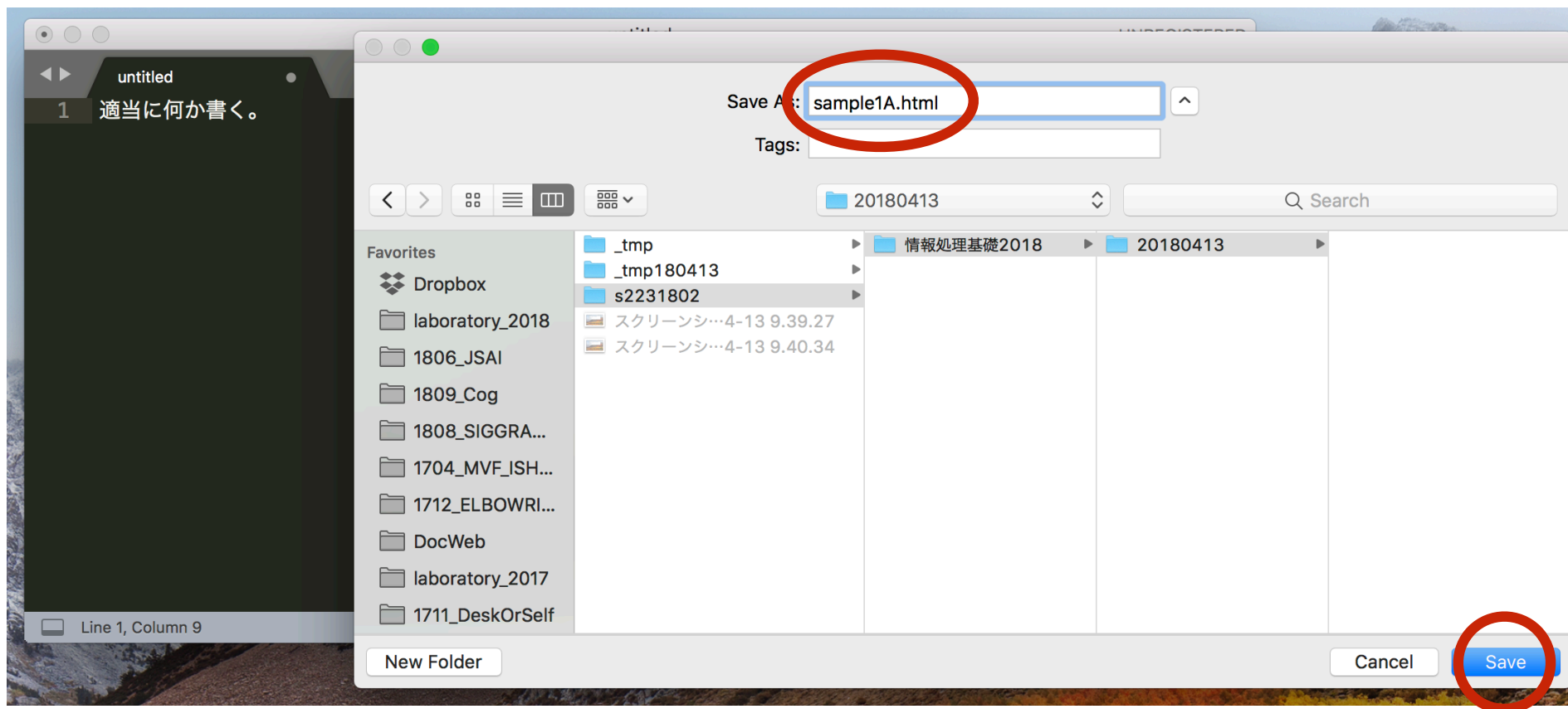


2) 文法モード (Syntax) をHTMLに設定



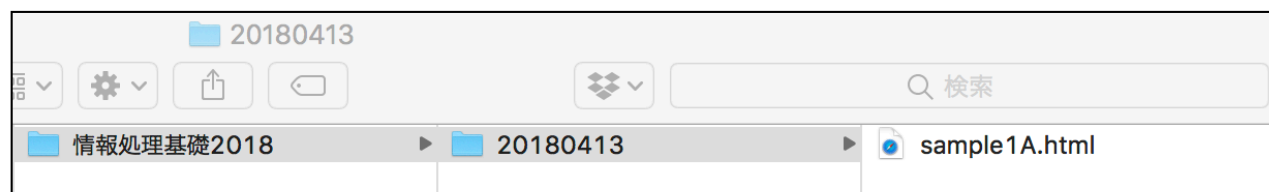
文書を保存する

3) 適当に何か書いた後で、 $\text{⌘}+S$ で保存します。



$\text{⌘}+S$

今回は、名前を「sample1A.html」として保存します。



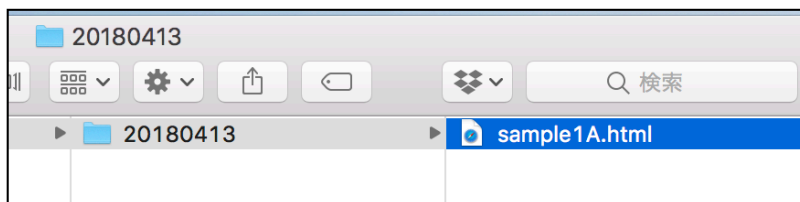
HTMLファイルの作成・保存・実行

以下のコードを入力したのち、 $\text{⌘}+S$ で保存します。

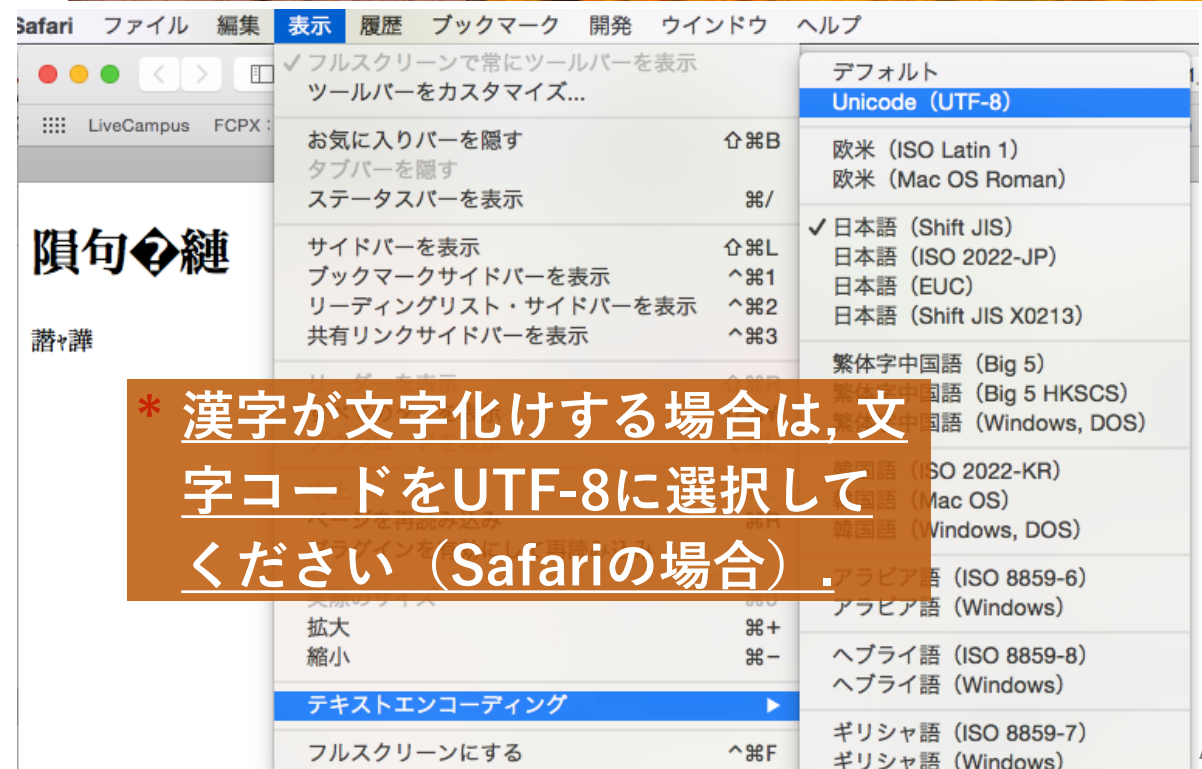
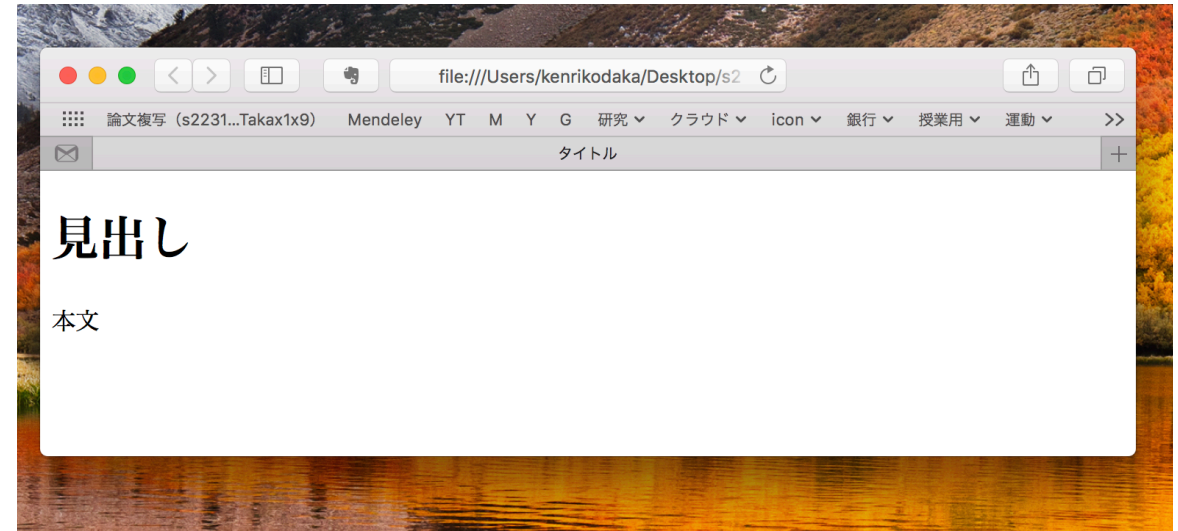
```
sample1A.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>タイトル</title>
5 </head>
6 <body>
7   <h1>見出し</h1>
8   <p>本文</p>
9 </body>
10 </html>
```

* $\text{⌘}+SHIFT+S$ で保存すると、別の名前で保存できます。

「sample1A.html」をダブルクリックして開きます。



ブラウザに以下のように表示されればOKです。



* 漢字が文字化けする場合は、文字コードをUTF-8に選択してください（Safariの場合）。

HTMLファイルの構造

sample1 A.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>タイトル</title>
5   </head>
6   <body>
7     <h1>見出し</h1>
8     <p>本文</p>
9   </body>
10 </html>
```

■ ドキュメントタイプ宣言
以下のテキストが、HTML文書であることを示すもの。

■ html要素
HTMLの実際のプログラムが記述される領域の指定。内部に、head要素とbody要素を1つずつ含む必要がある。

■ head要素
文書のメタ情報を記述する部分。ブラウザ上で表示されないことが多い。

■ body要素
文書の本体を記述する部分。

要素

<要素名> </要素名>

開始タグ

要素内容

終了タグ

要素の記述パターン

各要素の説明

タイトル要素

<title>タイトル</title>

- 文書のタイトルを指定. 本文の内容を間接的に表現する情報のため, 「head要素」の内部に置かれる. Safariでは, タイトルバーやタブの部分に表示される.

見出し要素

<h*>見出し</h*>

- 見出しの意味付けを与えるもの. *には1から6の数字が入り, 数字が低いほど, より高位で包括的な見出しであることを意味する.
- 右の文を書いて, 実行してみてください.

e.x. <h1>見出し 1 </h1>
<h2>見出し 2 </h2>
<h3>見出し 3 </h3>
<h4>見出し 4 </h4>
<h5>見出し 5 </h5>
<h6>見出し 6 </h6>

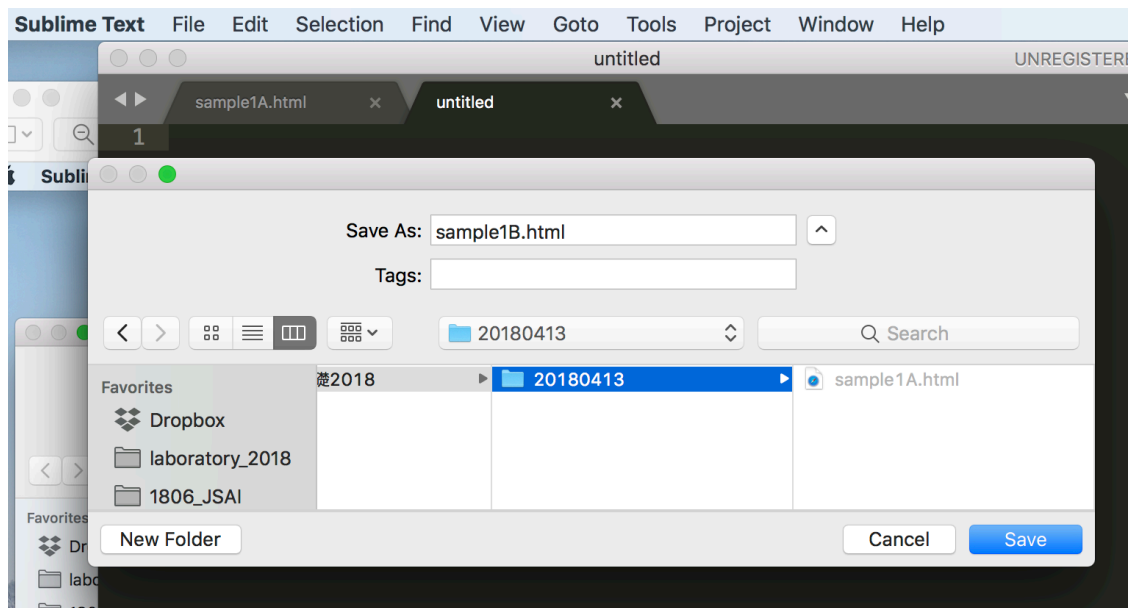
段落要素

<p>段落</p>

- pはparagraphの略です. すなわち, pタグで囲まれた文章は, 一つの段落を形成します.

新しいファイルを開く (sample1B.html)

⌘+Nで、新しいファイルを作成します。名前は、「sample1B.html」としましょう。



右のようなコードを入力し、保存してください。一回書いた歌詞は、コピー（⌘+N）とペースト（⌘+V）を使って複製していくと時間を節約できます。

sample1B.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>beatlesの歌詞</title>
6 </head>
7
8 <body>
9   <h1>A day in the life</h1>
10  <h2>初期状態</h2>
11  <p>
12    I read the news today oh boy.
13    About a lucy man who made the grade.
14  </p>
15  <h2>中央寄せ</h2>
16  <p align="center">
17    I read the news today oh boy.
18    About a lucy man who made the grade.
19  </p>
20  <h2>右寄せ</h2>
21  <p align="right">
22    I read the news today oh boy.
23    About a lucy man who made the grade.
24  </p>
25  <h2>改行</h2>
26  <p>
27    I read the news today oh boy. <br>
28    About a lucy man who made the grade.
29  </p>
30 </body>
31 </html>
```

改行の扱い（BR要素）

HTML文書の中の改行は、ブラウザでの表示にあたって実際の改行とは見なされないことに注意。

（BR要素）によって改行を実現。
BR要素の様に要素内容を持たない要素には終了タグは与えない。

整列属性（ALIGN）

P要素の開始タグの中に、段落の整列情報を指定することができる。

何も指定しない場合、デフォルトでは左寄せ（↓）として実行。

<p>



<p align="left">

beatlesの歌詞

file:///Users/kenri11/Dropbox/Docu リーダー

BBC Mendeley ko Yt h6YT Y G M

beatlesの歌詞

A day in the life

初期状態

I read the news today oh, boy. About a lucky man who made the grade. And though the news was rather sad.

中央寄せ

<p align="center">

I read the news today oh, boy. About a lucky man who made the grade. And though the news was rather sad.

右寄せ

<p align="right">

I read the news today oh, boy. About a lucky man who made the grade. And though the news was rather sad.

改行

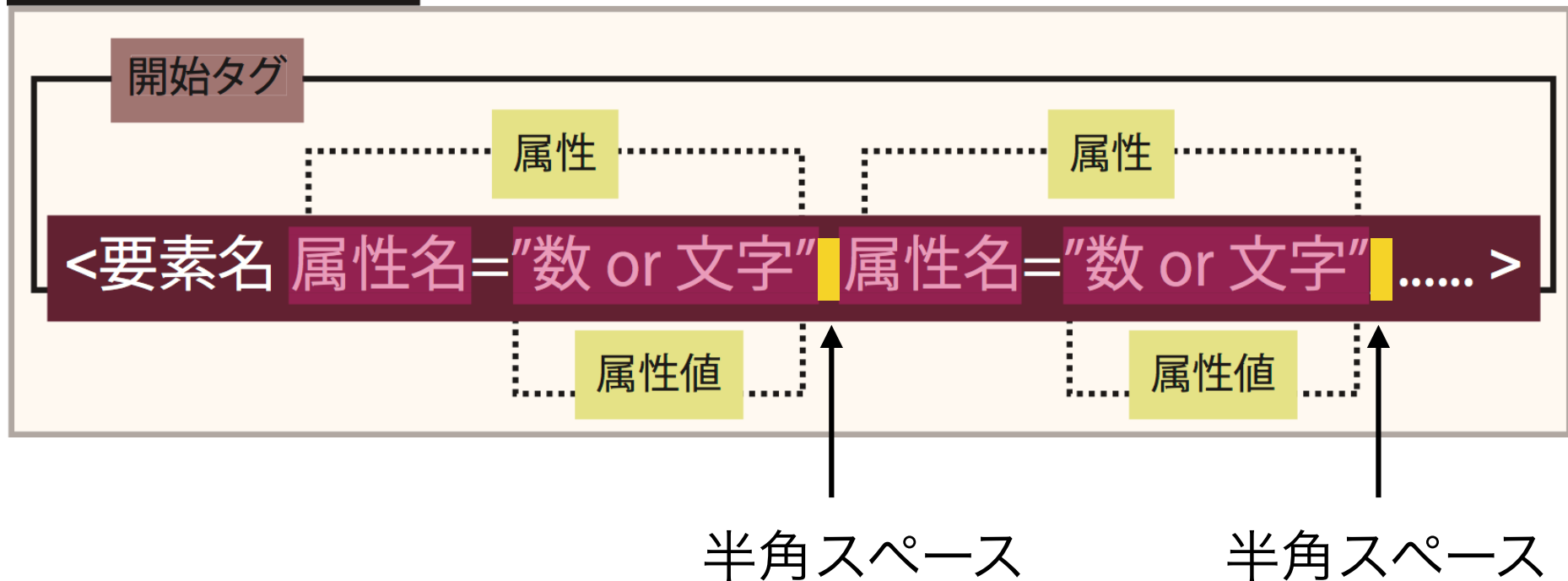
I read the news today oh, boy.
 About a lucky man who made the grade.
 And though the news was rather sad.

実行結果

属性の記法

属性は, HTMLの各要素に対して付加的な情報を与えるためのフォーマット. 開始タグに対して, 以下のようなパターンで記述.

図 - 属性の記述パターン



代表的な要素

a要素

```
<a href = "XXX", target = "YYY"> ... </a>
```

"XXX" リンク先のURL

"YYY" ウィンドウの指定方法

target = _blank

新しいウィンドウを開いて表示

target = _self

リンク元と同じウィンドウに表示

target = _parent

親ウィンドウに表示

target = _top

最上位ウィンドウに表示

```
<p>新しいページで <a href = "http://google.com", target = "_blank"> google </a> を開きます。</p>  
<p>現在のページで <a href = "http://google.com", target = "_self"> google </a> を開きます。</p>
```

ソース

代表的な要素

ul要素（リスト表記・番号無し）

```
<ul> <li>...</li> <li>...</li> ... </ul>
```

ol要素（リスト表記・番号付与）

```
<ol> <li>...</li> <li>...</li> ... </ol>
```

li要素の中に、リストの各要素を記述する。

```
<p>私が行ったことのある外国の都市を思い出してみます。</p>
<ul>
  <li>シンガポール</li>
  <li>ソウル</li>
  <li>ニース（フランス）</li>
  <li>ル・グラン＝ポルナン（フランス）</li>
  <li>セントポール（アメリカ）</li>
  <li>クアラルンプール（マレーシア）</li>
  <li>長春（中国）</li>
  <li>ハワイ</li>
  <li>シドニー</li>
  <li>台湾</li>
</ul>
```

私が行ったことのある外国の都市を思い出す。

- シンガポール
- ソウル
- ニース（フランス）
- ル・グラン＝ポルナン（フランス）
- セントポール（アメリカ）
- クアラルンプール（マレーシア）
- 長春（中国）
- ハワイ
- シドニー
- 台湾

ソース

実行例

代表的な要素

画像の表示：img要素

```

```

"XXX" 画像ファイル名 (URL)

"PPP" 画像の表示幅 (ピクセル)

"YYY" 画像を表すテキスト

"QQQ" 画像の表示高さ (ピクセル)

ソース

```
<p>画像表示の例 1 </p> ↓  
  
<p>画像表示の例2</p> ↓  
 ↓  
  
  

```

画像表示の例 1



画像表示の例2



実行結果

画像ファイルを指定する (パス)

 tmp.html

から

 kenrikodaka.jpg

を指定する方法

相対パス

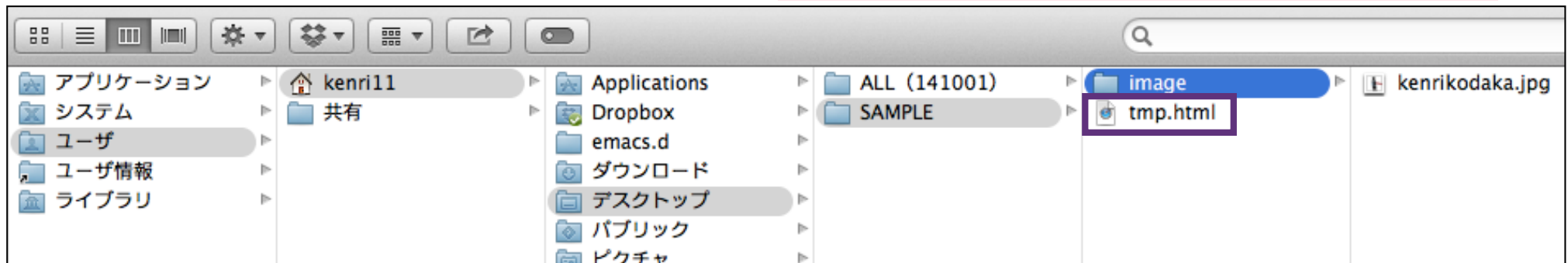
image/kenrikodaka.jpg

./image/kenrikodaka.jpg

絶対パス

/Users/kenri11/Desktop/SAMPLE/image/kenrikodaka.jpg

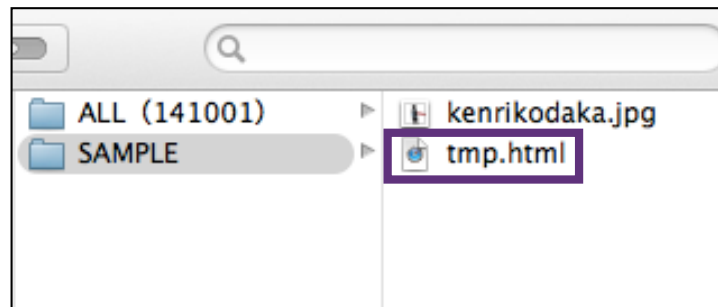
* terminalにドラッグすると絶対パスがわかります。



同じ階層にある場合

kenrikodaka.jpg

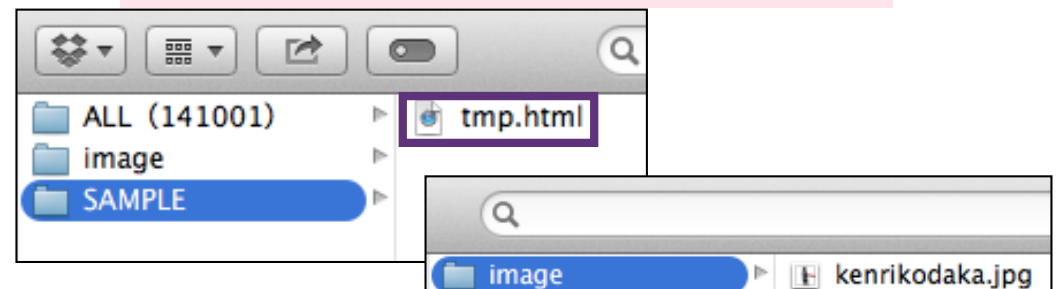
./kenrikodaka.jpg



別の階層にある場合

../image/kenrikodaka.jpg

* ../とすると、一つ上の階層に上がります。



コメントアウト

プログラムの中で、以下のような補足的な情報を書き込むことによって**保守性**が高まる。

作者・修正者・修正日 文に対する説明

* 「縦長にして画像を表示」「全ての段落を右寄で統一」など

チームで共有する場合の指示文

* 「この関数部分を修正してください」など

このような実際には実行されない文は**コメント文**と呼ばれ、プログラムの中で実行される文ではないことが明示的に示す必要がある。この操作を**コメントアウト**と呼ぶ。

HTMLのコメントアウト書式

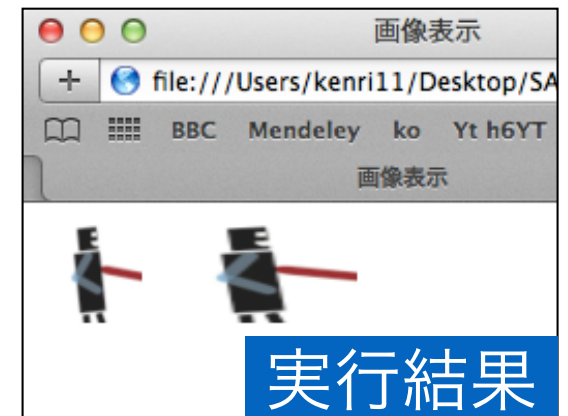
<!-- 実行されない文 -->

ソース

```
<body>↓
<!-- 通常比の図を表示 -->↓

<!-- 横長の図を表示 -->↓

</body>↓
```



実行結果

⌘ + /

* sublimeを含めた多くのエディタでは「⌘ + /」で、指定行が自動的にコメントに変換されます。

大文字と小文字の区別

HTML5では、要素の記述にあたって大文字と小文字の区別がない。各自で好みの表現を採用すればよい。

```
1 <!DOCTYPE html>↓
2 <HTML> ↓
3 <head>↓
4 <title>タイトル</title>↓
5 </HEAD>↓
6 <BODY>↓
7 <H1>見出し</h1>↓
8 <p>本文</P>↓
9 </BODY>↓
10 </html>
```

* 左のように、大文字と小文字が混在していても、問題なくトレースしてくれます。

自己紹介のホームページを作ってみよう.

- html要素・head要素・body要素・title要素は必須とします.
- body要素の内部では, 今回学んだ, h*要素・p要素・a要素・ul要素・img要素の全てを使用してください.
- ファイル名は, sample1X.htmlとします. 課題用のフォルダの中に, 学籍番号のフォルダを作って, その中に画像とともにコピーしてください.
- 必ず以下のコンテンツを含めてください.
 1. 使用しているPCで自分の写真を撮影し, img要素で表示.
 2. お気に入りのサイトを, a要素でリンク.
 3. 『私の好きな（嫌いな）XXX』（XXXは任意）というコンテンツを作って, 「ul要素」でリストを作成.
- 余裕のある人は, 別のタグを使ってコンテンツをより充実させてください.

HTML5の全要素のリファレンス

<http://www.html5.jp/tag/elements/>

補足1：表組み

table要素

tr要素

td要素

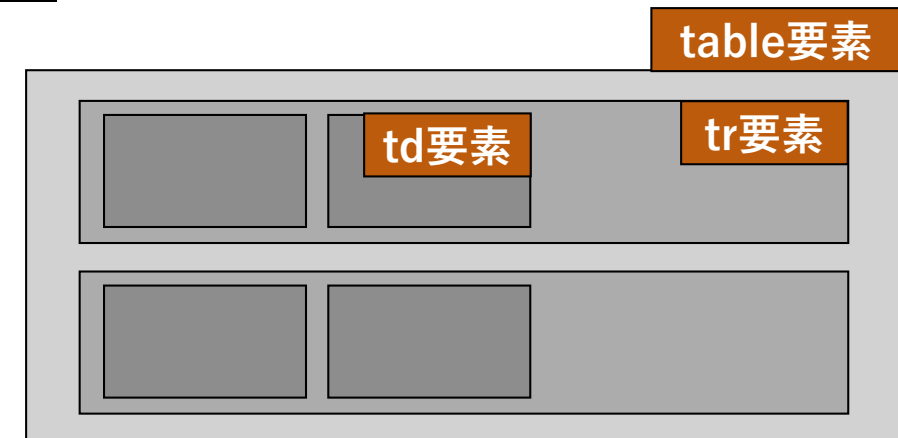
`<table border = "XXX", cellpadding = "YYY"> ... </table>`

"XXX" 枠線の太さ

"YYY" セル間の距離

```
<table border="1" cellpadding="2">
<tr>
  <td width="50" align="center">4/13</td>
  <td width="100">情報検索基礎</td>
  <td width="100">
    <strong>情報処理基礎</strong></td>
  <td>イントロダクション、HTML作成の基礎</td>
</tr>
<tr>
  <td>4/20</td>
  <td>情報検索基礎</td>
  <td><strong>情報処理基礎</strong></td>
  <td>HTMLファイル作成の基礎</td>
</tr>
<tr>
  <td>4/27</td>
  <td colspan="2" align="center">
    <strong>情報処理基礎</strong></td>
  <td>CSSの基礎（課題A）</td>
</tr>
</table>
```

ソース



`<td colspan = "XXX"> ... </td>`

`<td rowspan = "YYY"> ... </td>`

"XXX"

セルの結合数（行）

"YYY"

セルの結合数（縦）

実行結果

4/13	情報検索基礎	情報処理基礎	イントロダクション、HTML作成の基礎
4/20	情報検索基礎	情報処理基礎	HTMLファイル作成の基礎
4/27	情報処理基礎		CSSの基礎（課題A）

補足2：ページ内リンク

`<a href = "#XXX"> ... </a>`

`<要素 id = "XXX"> ... </要素>`

特定の要素に対して、idで名前を指定することで、a要素から#の頭文字を与えることで、ページ内にリンクを与えることができます。

```
1
2 <h1 id="mokuji">目次</h1>
3 <ul>
4   <li><a href="#part1">第1章</a></li>
5   <li><a href="#part2">第2章</a></li>
6   <li><a href="#part3">第3章</a></li>
7 </ul>
8 <hr>
9
10 <h1 id="part1">第1章</h1>
11 あいうえお
12 <hr>
13
14 <h1 id="part2">第2章</h1>
15 かきくけこ
16 <hr>
17
18 <h1 id="part3">第3章</h1>
19 さしすせそ
20 <hr>
21
22 <a href="#mokuji">戻る</a>
23
```

ソース

目次

- [第1章](#)
- [第2章](#)
- [第3章](#)

実行結果

第1章

あいうえお

第2章

かきくけこ

第3章

さしすせそ

[戻る](#)

hr 要素

`<hr>`

水平線を引きます。

補足3：図・キャプション

figure要素



img要素

figcaption要素

FIGURE 要素

IMG 要素

FIGCAPTION 要素

```
<p>
  小鷹研究室は、昨年度、
  Siggraph Asia2017（バンコク）において、
  二つのVRシステムを発表しました。
</p>
<p>
  Recursive Function Space [図1]は、
  幽体離脱を擬似的に体験することのできる装置です。
  Stretchar(m) [図2]は、
  あたかも自分自身の腕が伸びるような体験をすることが
  できる装置です。
</p>

<figure id="fig1">
  
  <figcaption> (図1) Recursive Function Space, 2017</figcaption>
</figure>

<figure id="fig2">
  
  <br>
  <figcaption> (図2) Stretchar(m), 2017</figcaption>
</figure>
```

ソース

小鷹研究室は、昨年度、Siggraph Asia2017（バンコク）において、二つのVRシステムを発表しました。

Recursive Function Space [図1]は、幽体離脱を擬似的に体験することのできる装置です。Stretchar(m) [図2]は、あたかも自分自身の腕が伸びるような体験をすることが出来る装置です。



(図1) Recursive Function Space, 2017 (小鷹・森)



(図2) Stretchar(m), 2017 (小鷹・森)

実行結果

補足4：youtubeを埋め込む

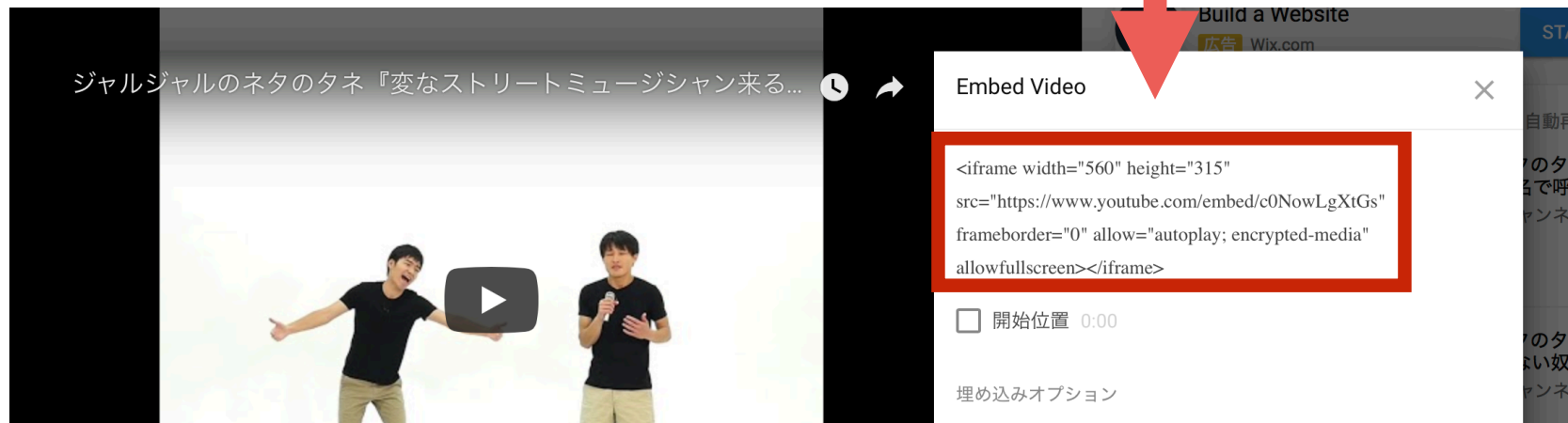


共有

<https://youtu.be/c0NowLgXtGs>

☐ 開始位置 0:14

埋め込む コピー



ジャルジャルのネタのタネ『変なストリートミュージシャン来ると思ってた変な奴』

視聴回数 75,369 回

共有

埋め込む

Embed Video

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/c0NowLgXtGs"
frameborder="0" allow="autoplay; encrypted-media"
allowfullscreen></iframe>
```

☐ 開始位置 0:00

埋め込みオプション

自動生成されたタグをコピーして、そのまま各自のHTMLファイルに貼り付ければ、自動的にYoutubeの窓が埋め込まれます。実際に再生するためには、インターネット環境が必要です。