

楽しいホームページの制作

科目の概要

みなさんはパソコンで、またはスマートフォンで、インターネットに日々アクセスしていることと思います。インターネット上には無数のホームページが存在していて、それらは「HTML（ハイパー・テキスト・マークアップ・ランゲージ）」と呼ばれる言葉で作られています。

この講義は、2014年に大幅改定された最新の言語「HTML5」を使って、スマートフォン向けのホームページを、実際に制作しながら理解を深めます。

学習の要点

本講義の内容は大きく3つです。

- ・HTMLマークアップ
- ・CSS3
- ・JavaScript

教材

- ・授業テキスト（印刷推奨）
- ・授業ページ（ブックマーク推奨）

<https://nakashimmer.github.io/YashimaHTML/>

- ・演習用のブラウザとしてGoogle Chrome を事前にインストール

目次

1. インターネットとWeb	2
2. テキストエディタ	3
3. HTMLの基本	4
4. CSSの基本	7
5. 様々なHTML	10
6. CSS セレクタとプロパティ	14
7. リッチなデザイン	15
8. 変形のCSS	16
9. CSSアニメーション	17
10. 擬似クラス	18
11. JavaScriptの基本	19
12. JavaScriptでコンテンツ操作	21
13. JavaScriptでCSS操作	22
14. Webアプリケーション	23
15. まとめ	24
16. 最終試験	24

講師：中島俊治(なかしま・しゅんじ)

岡山県生まれ、1997年上京ソフトバンク入社、GeoCities JAPAN立ち上げ、Yahoo! JAPAN、SBI、野村證券グループの後独立。現在は複数の大学やワークショップでHTML5をメインとした講座を開催。

一般社団法人 Web技術振興協会 代表理事
(<https://webtechpromo.org>)。

マイクロソフトMVPアワード受賞。

最終試験

- ・単位の認定は、出席率（原則100%）及び最終試験の結果を総合評価。
- ・最終試験は16コマ目。
- ・試験代替処置（レポート提出など）はない。
- ・最終試験は択一問題で、このテキストのほか、PDF資料、講師が講義した内容を含めた授業の中から出題。

1. インターネットとWeb

1.1. インターネットとWeb

- ・インターネットは、世界規模のネットワーク、それらの**情報通信技術**の総称。
- ・例 メール(SMTP、IMAP)、ファイル転送(FTP)、ホームページ閲覧(HTTP、HTTPS) 等

1.2. Web

- ・Webはインターネットのサービスのひとつ
- ・World Wide Web (www) の略。世界中に張り巡らせた「蜘蛛の巣」のたとえ
- ・Webに関連したもの：Webサイト・Webページ・Webサービス・「続きはWebで」・Webホスティング・Webアプリケーション

1.3. Webの特徴

- ・Hypertextが実装したシステム（リンクでつながっている。順番に関係なく閲覧できる）
- ・HTML、CSS、JavaScriptで主に記述
- ・技術の仕様はW3Cが策定する

1.4. Webの歴史

- ・1989年頃 CERN（欧州原子核研究所）のティム・バーナーズ・リー博士グループがWebシステムを作った。文字の表示とハイパーリンクを実装。
- ・1993年 NCSA が作ったMosaic ブラウザで画像を表示することが可能になった。
- ・1994年「NetScape Navigator」と「Internet Explorer」によるブラウザ戦争を教訓にWebの技術を標準化する団体「W3C」が設立された。
- ・1994年 Yahoo!開始。1996年 Yahoo! JAPAN開始・検索エンジンが主流に。
- ・2014年 HTML5がW3Cから勧告される。
- ・現在では、動画やアニメーションが可能になり、スマートフォンからのWeb閲覧が多くなった。
- ・2019年 WTATWG(Web Hypertext Application Working Group：ブラウザ業界団体)が開発を進め、W3Cは承認をする体制になった。

1.5. HTML5

- ・2014年10月28日に勧告された。
- ・1999年12月のHTML 4.01以来15年ぶりの大幅改訂。
- ・HTMLのタグのマークアップはページの構造とコンテンツの見栄えを定義していたが、HTML5では構造に特化して検索エンジンに内容データを的確に伝えることができるようになった。
- ・CSSは主にページの段組や色、背景、枠線などの見栄えを定義していたが、CSS3ではさらにリッチな見栄え、コンテンツの変形、アニメーション、3次元表示が可能になった。
- ・JavaScriptはブラウザ内での動きを定義することができるようになった。さらにAPI(Application Program Interface)により、様々な機能をJavaScriptで利用できるようになった。

1.6. ホームページ

- ・いろいろなホームページを見て行きましょう。

2. テキストエディタ

2.1. 演習環境について

本来練習に必要なものは次のふたつ

2.1.1. テキストエディタ(コードエディタ)でファイルを作成しコードを入力

- VisualStudio Code
- Atom
- DreamWeaver
- TeraPad(Win)
- mi(Mac)

2.1.2. 表示・動作を確認するのはブラウザ(今回はgoogle chrome)

2.2. 今回の講座について

ブラウザ(google chrome)は必須。

テキストエディタについては、入力が上手いかない方も想定し、本講座では簡易簡易テキストエディタを用意しました。

字下げはTABキーを使います。

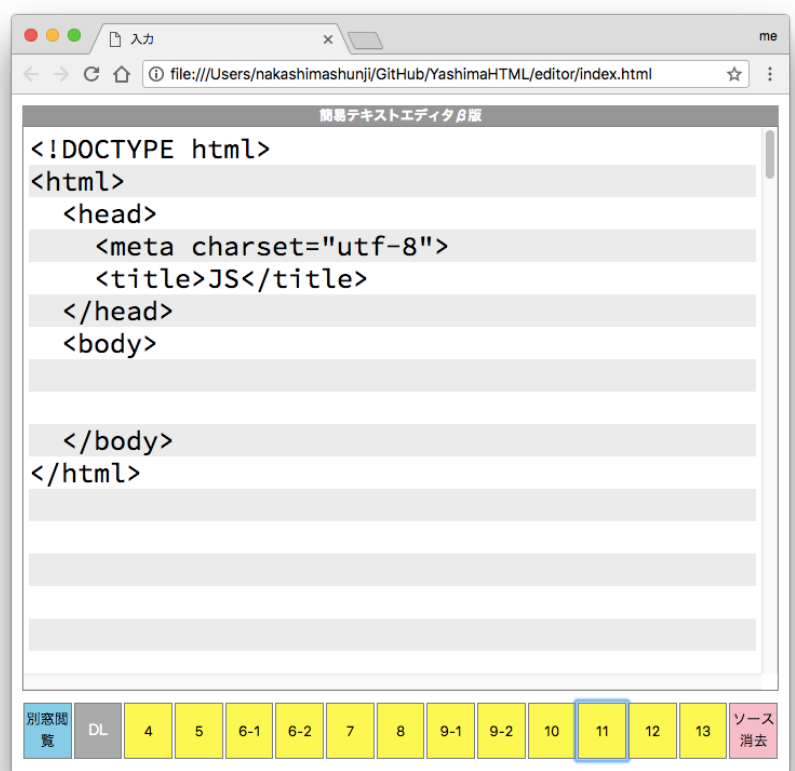
2.3. 今回の講座について

本講義では、まず講師が説明を行いますので、内容の理解につとめてください。…①

その後、皆さんがテキストエディタで実際にコードを書き、ブラウザで確認する演習することで知識の定着を行います。…②

①も②もできればいいのですが、初心者の方については、②の演習が思ったとおりに行かないこともあります。対面での授業ではないので間違いを一緒に探すことには限界があります。

ですので、まずは①の内容の理解を目指しましょう。



3. HTMLの基本

3.1. 記述

```
<p>きょうはとても良い天気です</p>
```

- <開始タグ> 内容 </終了タグ>
- p要素は段落の意味

3.2. ページの大まかな構造

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>

  </body>
</html>
```

- <!DOCTYPE html>は、この文書がHTML5のルールで書かれている
- html要素はHTMLのルート
- head要素はその文書の情報
- body要素はページに表示するコンテンツ

3.3. 各要素の詳細

3.3.1. head要素

```
<head>
  <meta charset="utf-8">
  <title>私の自己紹介</title>
  <meta name="viewport" content="width=device-width">
</head>
```

- head要素はその文書の情報
- meta要素は情報の情報
- <meta charset="utf-8">は文字コード
- title要素はページタイトル
- viewportは表示領域。

3.3.2. body要素

```
<body>
  <div id="wrap">
    <header>

    </header>
    <nav>

    </nav>
    <main>

    </main>
    <footer>

    </footer>
  </div>
</body>
```

- body要素はページに表示するコンテンツ。
- div要素は意味を持たない要素。ページの見栄えだけのために使う。
- header要素は文章のヘッダ(ページのタイトルやロゴなどのグループ)
- main要素はそこがメインコンテンツであることを示す。
- footer要素は文書のフッタ(固定リンクや著作権表示を含むグループ)を示す。

3.3.3. header要素

```
<header>
  <h1>私の自己紹介</h1>
</header>
```

- h1要素は大見出し

3.3.4. nav要素

```
<nav>
  <ul>
    <li><a href="index.html">ホーム</a></li>
    <li><a href="family.html">家族</a></li>
    <li><a href="contact.html">コンタクト</a></li>
  </ul>
</nav>
```

- nav要素はグローバルナビゲーション(メインリンク)
- ul、li要素は箇条書き
- a要素はリンク(hrefはリンク先)

3.3.5. main要素

```
<main>
  <section>
    <h2>ご挨拶</h2>
    <p>こんにちは〇〇です。</p>
  </section>

  <section>
    <h2>仕事</h2>
    <p>私の仕事は〇〇です。</p>
  </section>

  <section>
    <h2>趣味</h2>
    <p>私の趣味は〇〇です。</p>
  </section>
</main>
```

- main要素はメインコンテンツ
- section要素はひとつの内容のグループ
- h2要素は中見出し
- p要素は段落

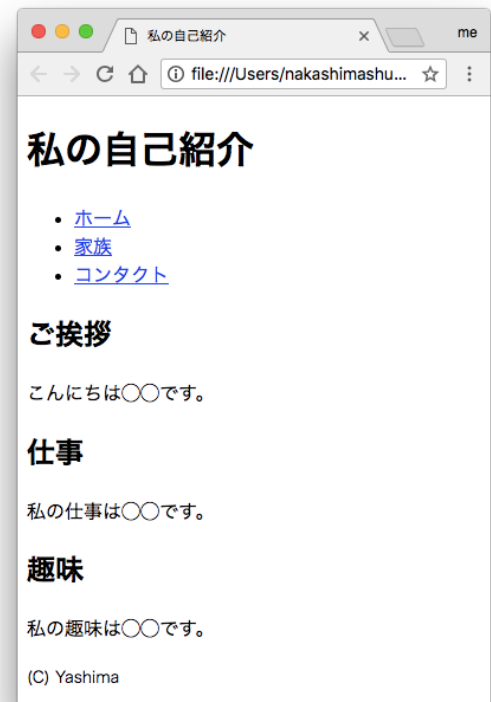
3.3.6. footer要素

```
<footer>
  <p><small>(C) Yashima</small></p>
</footer>
```

- footer要素は文書のフッタ(固定リンクや著作権表示を含むグループ)
- small要素は著作権

3.4. 参考：HTMLの全体は次の通り

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>私の自己紹介</title>
6.     <meta name="viewport" content="width=device-width">
7.   </head>
8.   <body>
9.     <div id="wrap">
10.      <header>
11.        <h1>私の自己紹介</h1>
12.      </header>
13.      <nav>
14.        <ul>
15.          <li><a href="index.html">ホーム</a></li>
16.          <li><a href="family.html">家族</a></li>
17.          <li><a href="contact.html">コンタクト</a></li>
18.        </ul>
19.      </nav>
20.      <main>
21.        <section>
22.          <h2>ご挨拶</h2>
23.          <p>こんにちは〇〇です。</p>
24.        </section>
25.
26.        <section>
27.          <h2>仕事</h2>
28.          <p>私の仕事は〇〇です。</p>
29.        </section>
30.
31.        <section>
32.          <h2>趣味</h2>
33.          <p>私の趣味は〇〇です。</p>
34.        </section>
35.
36.      </main>
37.      <footer>
38.        <p><small>(C) Yashima</small></p>
39.      </footer>
40.    </div>
41.  </body>
42.</html>
```



4. CSSの基本

4.1. CSSの記述

```
<head>
  <meta charset="utf-8">
  <title>私の自己紹介</title>
  <meta name="viewport" content="width=device-width">
  <style>
    p{color:skyblue;}
  </style>
</head>
```

- CSSは、**style要素**の中に記述する
- 書式は、**セクタ {プロパティ: 値;}**
- colorプロパティは文字の色

4.2. CSSを記述する

一旦p要素へのCSSを削除し、次の内容をstyle要素内に追加してください。

4.2.1. リセットCSS

```
*{
  margin:0px;padding:0px;
  font-size:16px;
}
```

- *ユニバーサルセクタ
- font-size: 文字の大きさ

4.2.2. ページ全体

```
#wrap{
  width:320px;
  margin:0px auto;
  border:1px solid gray;
}
```

- width: 横幅
- margin: 外の余白
- border: 枠線

4.2.3. ヘッダー

```
header h1{
  text-align:center;
  line-height:2em;
  font-size:1.5em;
  background-color:#eee;
}
```

- text-align: 文字揃え
- line-height: 行の高さ
- background-color: 背景色

4.2.4. ナビゲーション (横並べ)

```
nav ul{
  display:flex;
  list-style:none;
}
nav li{
  flex:1;
  text-align:center;
  border:1px solid gray;
  margin:2px;
}
```

- display: 表示の状態 (親要素に display:flex;、子要素に flex:1; とすると子要素は横に並ぶ)
- list-style: 箇条書きの行頭記号 (noneは記号なし)

4.2.5. セクション

```
section{  
  border:1px solid gray;  
  margin:2px;  
}
```

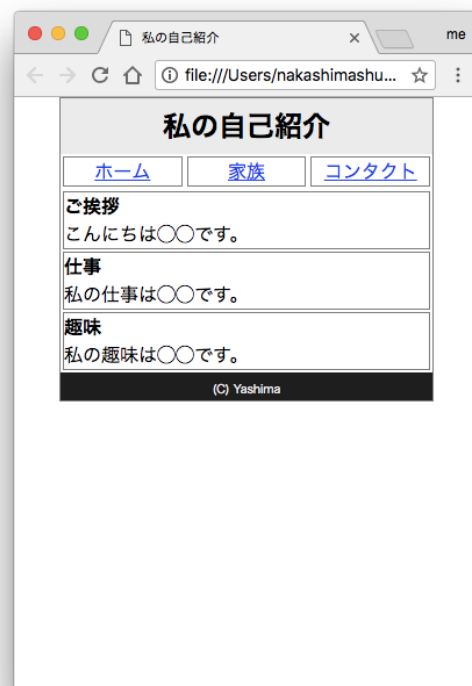
4.2.6. フッター

```
footer{  
  text-align:center;  
  background-color:#222;  
  color:#eee;  
}  
footer small{  
  font-size:0.5em;  
}
```

- em：大きさの単位（1文字分）
- #222：色の値

4.3. 参考：CSSの全体は次の通り

```
1. <style>
2. *{
3.   margin:0px;padding:0px;
4.   font-size:16px;
5. }
6. #wrap{
7.   width:320px;
8.   margin:0px auto;
9.   border:1px solid gray;
10.}
11.header h1{
12.  text-align:center;
13.  line-height:2em;
14.  font-size:1.5em;
15.  background-color:#eee;
16.}
17.nav ul{
18.  display:flex;
19.  list-style:none;
20.}
21.nav li{
22.  flex:1;
23.  text-align:center;
24.  border:1px solid gray;
25.  margin:2px;
26.}
27.section{
28.  border:1px solid gray;
29.  margin:2px;
30.}
31.footer{
32.  text-align:center;
33.  background-color:#222;
34.  color:#eee;
35.}
36.footer small{
37.  font-size:0.5em;
38.}
39.</style>
```



5. 様々なHTML

5.1. テキストのHTML

5.1.1. 強調・重要

```
<p>きょうは<em>いい天気</em>です</p>

<p><strong>HTML5は重要！</strong></p>
```

5.1.2. 引用

```
<p>クラーク博士は言いました。<q>少年よ、大志を抱け</q>と</p>

<p>以下引用します</p>
<blockquote>
  この講義は、2014年に大幅改定された最新の言語「HTML5」を使って、
  スマートフォン向けのホームページを実際に制作しながら理解を深めます。
</blockquote>
```

5.1.3. ルビ

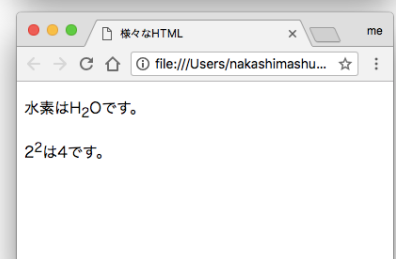
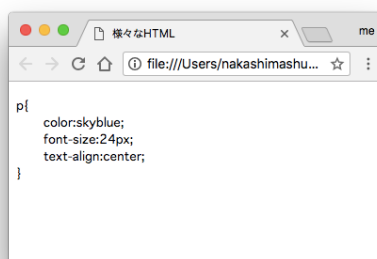
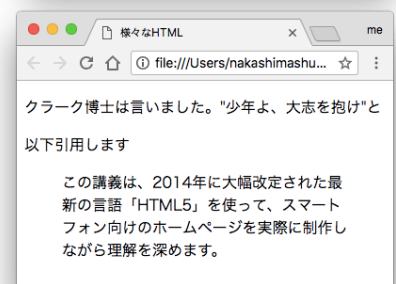
```
<p>今週は
  <ruby>
    <rb>大阪</rb>
    <rt>おおさか</rt>
  </ruby>
  に出張です。</p>
```

5.1.4. コード・整形済みテキスト

```
<p><pre><code>p{
  color:skyblue;
  font-size:24px;
  text-align:center;
}</code></pre></p>
```

5.1.5. 添字

```
<p>水素はH<sub>2</sub>Oです。</p>
<p>2<sup>2</sup>は4です。</p>
```



5.2. 映像と音声

5.2.1. 映像の要素

```
<p>
  <video src="video.mp4" controls
autoplay loop width="314"></
video>
</p>
```

- video要素は映像の要素。
- controls属性を付け加えるとコントロールが表示される。
- autoplay属性を付け加えると自動実行する。
- loop属性を付け加えると無限に繰り返し再生する。
- スマホの場合は、オーディオプレイヤーが表示される。

5.2.2. 音声の要素

```
<p>
  <audio src="audio.mp3" controls
autoplay loop></audio>
</p>
```

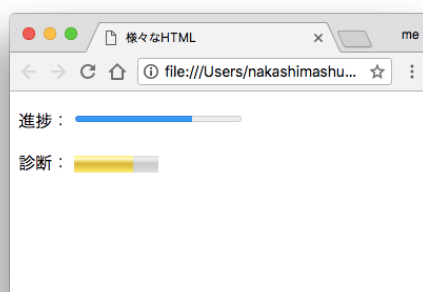
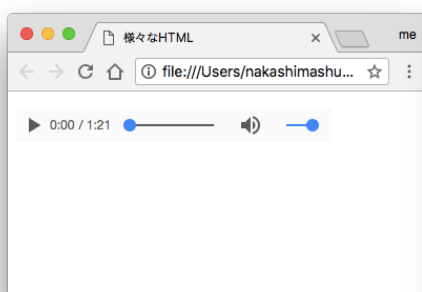
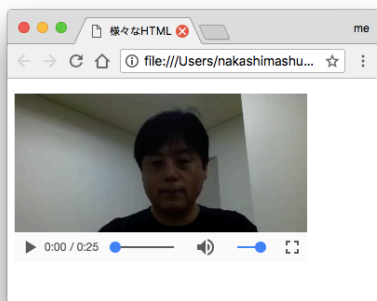
- audio要素は音声の要素。スマホの場合は、オーディオプレイヤーが表示される。属性はaudioと同じ。

5.3. 進捗や診断の部品

文字や数字の表示だけではなく、進捗や診断のバーを表示することができます。

```
<p>進捗：
  <progress value="70" max="100">70%</progress>
</p>
<p>診断：
  <meter value="70" min="0" low="50" high="60"
optimum="55" max="100">70kg</meter>
</p>
```

- progress要素は進捗状態を、meter要素は診断を表す
- value属性は値
- max属性は最大値、min属性は最小値
- low属性は最小の基準値、high要素は最大の基準値
- optimum属性は最適値



5.4. ボタン

アプリケーション用のボタン要素があります。

```
<p>
  <button onclick="alert('ボタン1を押しました');">ボタン1</button>
</p>
```

- button要素はボタンを表す。
- onclickはボタンをクリックしたら動作するプログラム名を示す。
- alert()はアラートを表示する。

5.5. 入力

アプリケーション用の入力ボックスなどが増えました。

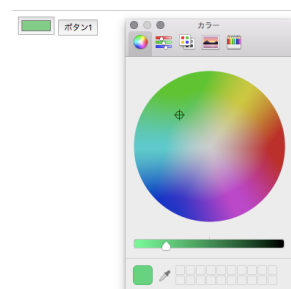
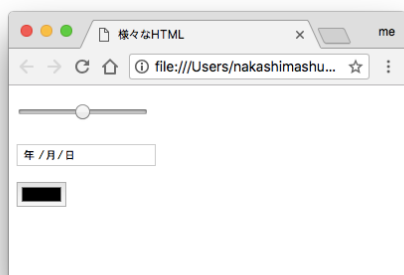
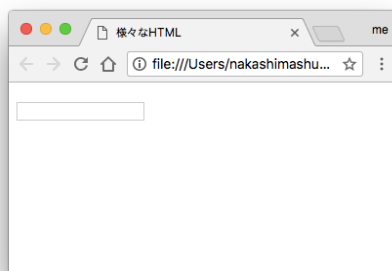
```
<p>
  <input type="text" id="txt1">
</p>
```

- input要素は入力ボックス
- type属性にtextを指定するとデフォルトの文字入力ボックスになる。
- id属性はその部品の固有の名前

5.6. 入力の様々な部品

```
<input type="range">
<input type="date">
<input type="color">
```

- rangeはスライダ、dateはカレンダー、colorはカラーピッカーを表示する

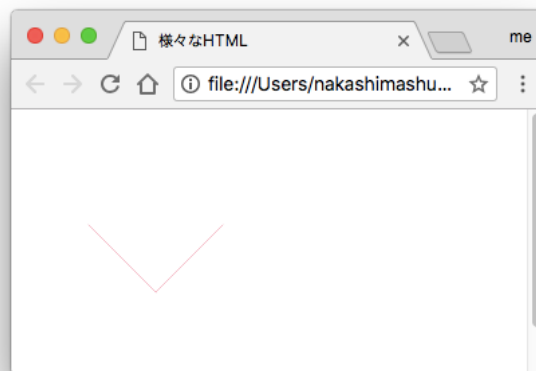


5.7. 描画の要素

```
<p>
  <canvas id="canvas1" width="200" height="200"></canvas>
</p>

<script>
  var canvas1 = document.getElementById("canvas1");
  var ctx1=canvas1.getContext("2d");
  ctx1.strokeStyle="pink";
  ctx1.moveTo(50,70);
  ctx1.lineTo(100,120);
  ctx1.lineTo(150,70);
  ctx1.lineWidth=1;
  ctx1.lineCap="round";
  ctx1.stroke();
</script>
```

- script要素はJavaScriptのグループであることを示す。JavaScriptでは大文字小文字は厳格なので注意。続きは、楽しいWebアニメーションで講義する



6. CSS セレクタとプロパティ

6.1. セレクタ

```
<style>
  p{color:skyblue;}
  #id1{color:red;}
  .class1{color:green;}
</style>

<body>
  <p>コンテンツAです。</p>
  <p class="class1" id="id1">コンテンツB</p>
  <p class="class1">コンテンツCです。</p>
  <p class="class1">コンテンツDです。</p>
  <p class="class1">コンテンツEです。</p>
</body>
```

セレクタの種類を覚えましょう。

- セレクタには、要素セレクタ、idセレクタ、classセレクタなどがあります。
- ユニバーサルセレクタ(*:全ての要素)もあります。



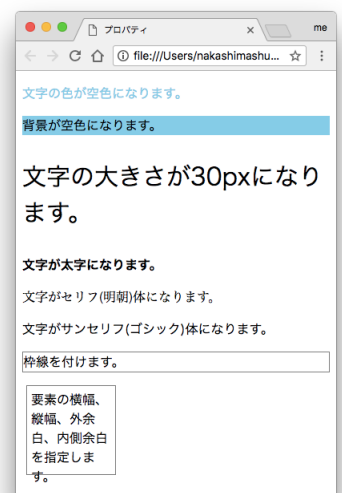
6.2. プロパティ

6.2.1. CSSを適用するHTMLは次のとおり

```
<body>
  <p class="css1">文字の色が空色になります。</p>
  <p class="css2">背景が空色になります。</p>
  <p class="css3">文字の大きさが30pxになります。</p>
  <p class="css4">文字が太字になります。</p>
  <p class="css5">文字がセリフ(明朝)体になります。</p>
  <p class="css6">文字がサンセリフ(ゴシック)体になります。</p>
  <p class="css7">枠線を付けます。</p>
  <p class="css8 css7">要素の横幅、縦幅、外余白、内側余白を指定します。</p>
</body>
```

```
<style>
  .css1{color:skyblue;}
  .css2{background-color:skyblue;}
  .css3{font-size:30px;}
  .css4{font-weight:bold;}
  .css5{font-family:serif;}
  .css6{font-family:sans-serif;}
  .css7{border:1px solid gray;}
  .css8{
    width:100px;height:100px;
    margin:5px;padding:5px;
  }
</style>
```

- colorは文字の色
- background-colorは背景色
- font-sizeは文字の大きさ
- font-weightは文字の太さ
- font-familyは文字の種別。
- borderは枠線
- widthは横幅
- heightは縦幅。
- marginは外側余白、
- paddingは内側余白



7. リッチなデザイン

7.1. リッチなデザイン

今までできなかった、影や角丸、グラデーションが簡単にできるようになりました。

```
<style>
  div{
    width:200px;height:200px;
    margin:50px auto;
    line-height:200px;
    text-align:center;
    border:1px solid gray;
  }
</style>

<body>
  <div id="box1">リッチなデザイン</div>
</body>
```

- テンプレートのファイルを用意



7.2. 背景画像のプロパティ

```
#box1{
  background-image:url(image.png);
  background-size:100% 100%;
}
```

- background-imageは、背景画像
- background-sizeは、背景画像の大きさ

7.3. 影のプロパティ

```
#box1{
  box-shadow:3px 3px 3px gray;
}
```

- box-shadowプロパティは、影をつける。

7.4. 角丸のプロパティ

```
#box1{
  border-radius:50px;
}
```

- border-radiusプロパティは、枠線(border)の角を丸くする。
- radiusは半径のこと

7.5. グラデーションのプロパティ

```
#box1{
  background:linear-gradient(white,blue);
}
```

- linearは徐々にという意味。



8. 変形のCSS

8.1. 変形のプロパティ

```
<style>
  div{
    width:200px;height:200px;
    margin:0 auto;
    line-height:200px;
    text-align:center;
    border:1px solid gray;
  }
</style>

<body>
  <div id="box1">変形のCSS</div>
</body>
```

さらに、回転や伸縮、歪みや移動もできるようになりました。

まず、左のようなソースがあるとし

8.1.1. 回転

```
#box1{
  transform:rotate(20deg);
}
```

- transformプロパティは、変形を指定する。
- 値のrotate()は角度を設定する。

8.1.2. 伸縮

```
#box1{
  transform:scale(2,1);
}
```

- 値のscale()は横倍率、縦倍率を設定する。

8.1.3. 移動

```
#box1{
  transform:translate(200px,0px);
}
```

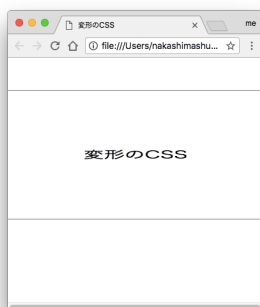
- 値のtranslate()は横の移動距離、縦の移動距離を設定する。

8.1.4. 変形の合成

transformプロパティの値として、半角空白で並べることができます。

```
#box1{
  transform:translate(200px,0px) rotate(30deg) scale(2,1);
}
```

- 横に200px移動し、30度回転し、横に2倍の変形になる。



9. CSSアニメーション

9.1. トランジション

9.1.1. 変形をトランジション

```
#box1{transition:transform 1s linear;}
#box1:hover{
  transform:translate(200px,0px) rotate(30deg);
}
```

- transitionは時間をかけて変化する
- hover:マウスがのったら、そのCSSを実行する。

9.1.2. CSSをトランジション

```
#box1{transition:color 1s linear;}
#box1:hover{color:red;}
```

- transitionプロパティは変形プロパティ以外にも取付可。

9.1.3. 複数のトランジション

```
#box1{transition:all 1s linear;}
#box1:hover{
  color:white;
  background-color:green;
  width:300px;
}
```

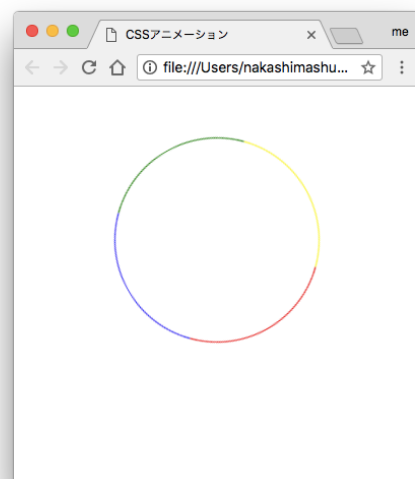
- allにまとめてしまいます。

9.2. アニメーション

9.2.1. スピナー

```
#box1{
  border-color:red blue green yellow;
  border-radius:50%;
  animation:spinner1 3s linear infinite normal;
}
@keyframes spinner1{
  100%{transform:rotate(360deg);}
}
```

- allにまとめてしまいます。



10. 擬似クラス

10.1. 擬似クラス

```
<ul>
  <li>項目1</li>
  <li>項目2</li>
  <li>項目3</li>
  <li>項目4</li>
  <li>項目5</li>
  <li>項目6</li>
  <li>項目7</li>
  <li>項目8</li>
  <li>項目9</li>
  <li>項目10</li>
</ul>
```

- 最初と最後の要素を、要素にclassなどの属性を付けずに指定します。たとえばこのようなソースがあるとする。

10.1.1. 最初と最後の要素

```
li:first-of-type{background-color:pink;}
li:last-of-type{background-color:skyblue;}
```

10.1.2. 奇数と偶数の要素

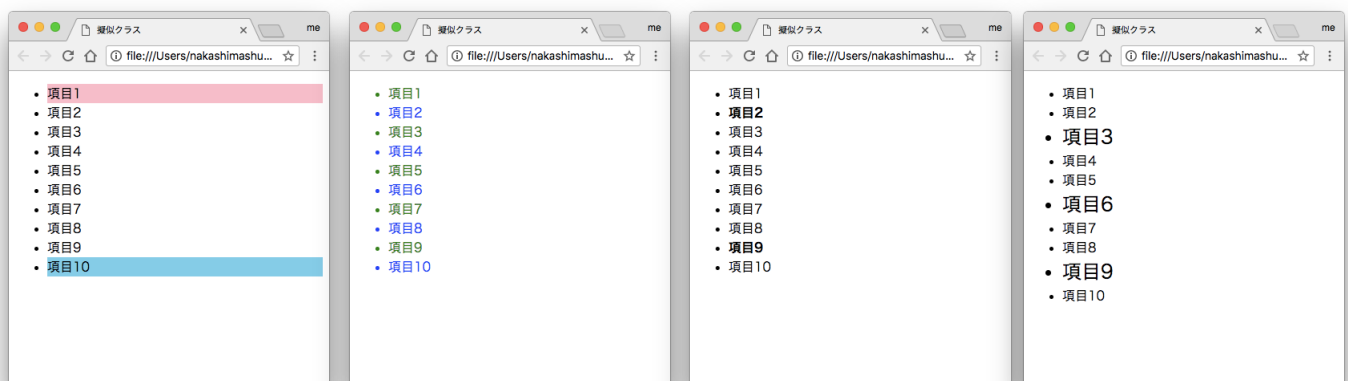
```
li:nth-of-type(odd){color:green;}
li:nth-of-type(even){color:blue;}
```

10.1.3. 上から、下から何番目

```
li:nth-of-type(2){font-weight:bold;}
li:nth-last-of-type(2){font-weight:bold;}
```

10.1.4. 何番目ごと

```
li:nth-of-type(3n){font-size:1.5em;}
```



11. JavaScriptの基本

11.1. JavaScript

JavaScriptは「プログラム」と考えるとわかりやすいでしょう。

ページへの表示、入力ボックスからの値の取得、計算、日付の取得は勿論、APIの力を借りて、GPSで現在地がわかり、シンセサイザを作ることでもでき、音声を出したり理解したり、ゲームを作ることでもできます。

11.1.1.記述の方法

```
<script>
  var a = 1;
  var b = 2;
  var c = a+b;
  document.write(c);
</script>
```

- `var`はその次のもの(例えば`a,b,c`など)が変数(variable)という箱を新しく生成する。
- `document.write()`は、ページに表示をする。

11.1.2.文字列の連結

```
<script>
  var a = "八洲";
  var b = "学園";
  var c = "大学";
  var d = a+b+c;
  document.write(d);
</script>
```

- `+`の記号は、文字列に関しては連結になる。

11.1.3.if文

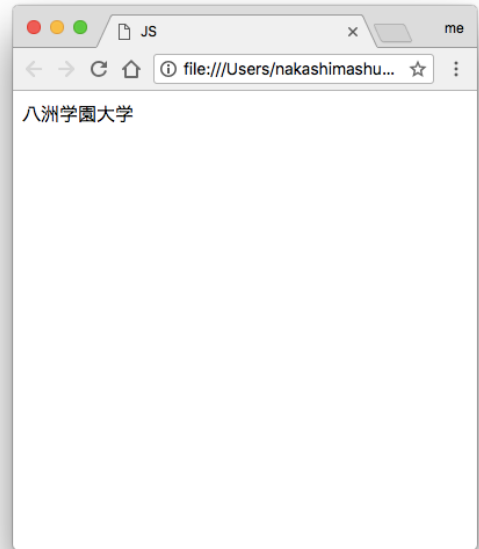
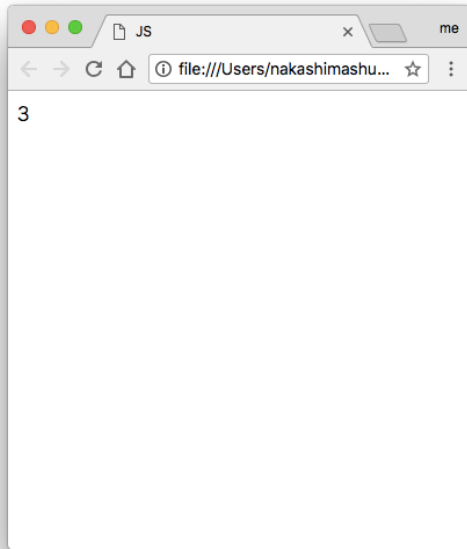
```
<script>
  var a=1;
  var b=2;
  if(a === b){var msg = "aもbも等しい";}
  if(a < b){var msg = "bが大きい";}
  if(b < a){var msg = "aが大きい";}
  document.write(msg);
</script>
```

- `if(条件式){処理A}else{条件を満たさなかった場合の処理B}`
- 等しいかどうかの条件式では、「`===`」を使う。(JavaScriptでは「`=`」は代入の意味)

11.1.4.乱数

```
<script>
  var a = 1;
  var b = Math.floor(Math.random()*3);
  if(a === b){var msg = "aもbも等しい";}
  if(a < b){var msg = "bが大きい";}
  if(b < a){var msg = "aが大きい";}
  document.write(msg);
</script>
```

- `Math.random()`メソッドは、0以上1未満の小数を毎回ランダムに取得する。
- `Math.floor()`メソッドは、小数点以下を切り捨てる。
- `math.floor(Math.random())`は、「0,1,2」のどれかを毎回ランダムに取得する。



12. JavaScriptでコンテンツ操作

12.1. 要素の取得

今までもでてきましたが、JavaScriptでは、要素の取得というものが一番大事です。

```
<p id="output1">ここが出力です</p>
```

```
<script>
  var output1=document.getElementById("output1");
</script>
```

- JavaScriptはscript要素の中にいるので、外にある他の要素に勝手にアクセスができない。
- ページを操作している「document」というものに、id名を示してその要素を取得してもらう。

12.2. 出力

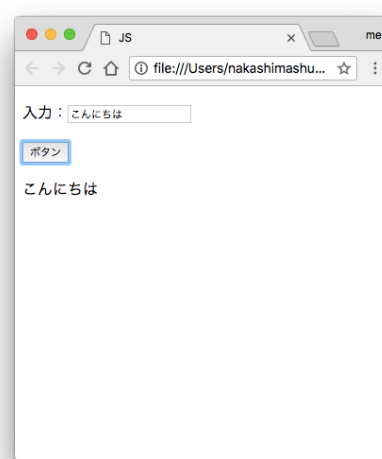
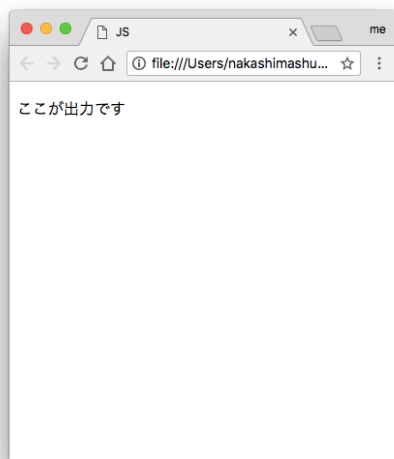
```
<p id="output1">ここが出力です</p>
<p><button onclick="btn1();">ボタン</button></p>
<script>
  var output1 = document.getElementById("output1");
  function btn1(){
    output1.innerHTML = "書き換えました";
  }
</script>
```

- ページ上に、値や文字を表示させることができます。
- innerHTML は、要素内のHTMLを上書きする。

12.3. 入力

```
<p>入力:<input type="text" id="input1"></p>
<p><button onclick="btn1();">ボタン</button></p>
<p id="output1">ここが出力です</p>
<script>
  var input1=document.getElementById("input1");
  var output1 = document.getElementById("output1");
  function btn1(){
    output1.innerHTML = input1.value;
  }
</script>
```

- 外の要素にアクセスできれば、値を取得することができます。
- value は、input要素など入力の部品に入力された値(文字列)を指す。上記の場合、入察された値をn1に代入している。



13. JavaScriptでCSS操作

13.1. 基本的な記述

先の回の要素の取得を使い、styleプロパティに値を設定します。

```
<div id="box1">BOX1</div>
<button onclick="btn1();">ボタン</button>

<script>
  var box1 = document.getElementById("box1");

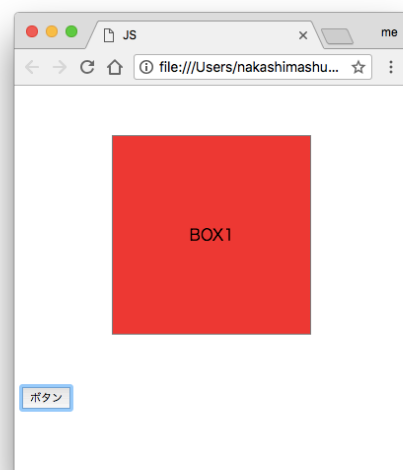
  function btn1(){
    box1.style.color="red";
  }
</script>
```

- `txt1.style.color="red"`とCSSの`#txt1{color:blue;}`は同じ働きです。
- CSSは書き換えできませんが、JavaScriptでは都度書き換えが可能です

13.2. ハイフンのついたCSSプロパティの扱い

```
function btn1(){
  box1.style.background-color="green";
}
```

- 左のコードは動かない。
- JavaScriptがプログラムなので「-」は使えない。
- 「-」を消し、次の単語の先頭を大文字にする。



14.Webアプリケーション

14.1.Webアプリケーション

私たちは、ホームページの制作のために、HTMLを学び始めましたが、かなり方向が違うように感じられたかと思います。

実は、今までのHTMLは確かにホームページのための技術でしたが、2014年になって、HTML5へと改定されてから、**Webアプリケーションを作るための技術**に大幅に改訂されました。

そのため、JavaScriptというプログラムが、とてもウエイトを占め、デザインのCSSやコンテンツのHTMLをコントロールするようになりました。

14.2.データストレージ

例えば、ゲームをして、明日になって続きをしようと思ったら、獲得したアイテムや得点などの情報は、サーバに送って保管するなどしか方法はありませんでした。

しかし、HTML5では、ブラウザを終了させても、**ブラウザの中にデータを保存する仕組み**がその回っています。WebStorageといい、LocalStorageとSessionStorageの2つがあります。またデータベースの実装も進んでいます。

14.3.スマホアプリ

皆さんは、スマホにアプリをインストールしているでしょう。

開発するための言語は、Android端末でしたらJava、iPhoneでしたらObjective-CやSwiftです。しかしひとつのアプリを公開するのに2倍のコードを書かなくてはなりません。

しかし、HTML5でコードを書き、スマホ用のアプリに変換してくれるクラウドサービスがあります。これを使えば、AppStoreやGooglePlayで配布することも可能です。

これをハイブリッドアプリケーションと呼びます。

14.4.WoT (IoT)

インターネットとものをつなぐことをIoT(Internet of Thing)といいます。その中でHTML5の技術でもものをつなげることをWoT(Web of Thing)といいます。

本来、Webブラウザの中で動作するWeb技術がブラウザでないところで活躍を始めています。

14.5.API

Application Program Interface は、わざわざ最初から開発しなくても、必要な機能を利用させてくれる仕組みです。外部サイトで提供しているAPIとHTML5の内部で持っているAPIと2種類あります。

14.5.1.GeoLocation API

HTML5は、ブラウザを通じてGPSの情報を取得することができます。

このAPIをGeoLocation APIといい、Googleの提供している地図情報と組み合わせることで、地図アプリを作ることが可能です。

14.5.2.Web Audio API

HTML5は、シンセサイザにもなります。ここでは、Web Audio APIを使っています。音源から波形の異なる音を作ることができます。音階も変えられます。

14.5.3.Audio API Video API

私たちはaudio要素、video要素を用いましたが、JavaScriptで速さや音の大きさなどをコントロールすることができます。これらをAudio API、Video APIと言います。

14.5.4.SpeechRecognition API

音声の仕組みがインストールされているPCでしたら、入力したテキストを元に音声を出力することができます。また、音声を認識させることや、日本語、英語の別も可能です。

14.5.5.Canvas API

Canvas APIはHTML5普及の立役者です。**描画**の機能やアニメーション、そしてゲームやユーティリティアプリを作ることが可能になりました

15.まとめ

- 本科目のまとめを行います。

16.最終試験

- 単位の認定は、出席率（原則100%）及び最終試験の結果を総合評価します。
- 試験は15回終了後、引き続き行われます。
- 試験は1時間です。
- 最終試験は択一問題で、このテキストを含み授業の中から出題されます。
- 試験の代替処置（レポート提出等）はありません。