

HTML 入門

緑川章一

HTML とは、HyperText Markup Language の略である。マークアップ言語 (Markup Language) とは、タグ (< と > によって囲まれた表現方法の指示) によって構造化された言語である。

1 基本構造

メモ帳を開いて、以下の sample0.html の内容を記述する。拡張子 (ドット. の後) は、html とする。ファイルを閉じたあと、プログラムを修正する場合には、アイコンを左クリックし、「プログラムから開く」の中のメモ帳を選択する。

```
sample0.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title></title>
<script>
    // ここには、JavaScript を記述します。
</script>
</head>
<body>

    <!--
        ここには、本文を記述しますが、
        これは、表示されません。
    -->

    これは、表示されます。

</body>
</html>
```

ファイルを保存するときには、左上隅の「ファイル」をクリックし、その中の「名前を付けて保存」を選択します。そこで、「ファイル名」は sample0.html, 「ファイルの種類」は、すべてのファイル (*.*) を選択します。一番下の「文字コード」は、UTF-8 を選択します。

このファイル (sample0.html) をダブルクリックで開くと、「これは、表示されます。」だけが表示されます。

【注】

- (1) <html>, </html>, <head>, </head>, <script>, </script> などのように、鍵カッコ <> で囲まれたものをタグ (荷札) と言う。タグは開始タグと終了タグ (/が付いている) でセットになっている。
- (2) <!DOCTYPE html> で文書が HTML5 の規格であることを宣言する。
- (3) <html> ~ </html> で囲まれた内容が HTML 文書であることを宣言する。
- (4) <html lang="ja"> html 文書は日本語 (ja) で書かれていることを表す。
- (5) <head> ~ </head> には、文字コードやタイトル、JavaScript や CSS の読み込みなど、本文に表されない文書自身に関するメタ情報を書く。
- (6) <meta charset="UTF-8"> 文字コードを記述する。ここでは、UTF-8 を用いる。文字コードとしては、shift_jis などとも用いられるが、現在では、UTF-8 が世界標準となっている。
- (7) <title> ~ </title> にタイトル (題名) を書く。
- (8) <script> ~ </script> には、必要ならば、JavaScript で命令を記述する。もちろん、何も書かなくとも良い。プログラムの機能や使用方法についてのコメント (注釈) を書く場合には、//から始める。//以降の文章は、プログラムとして実行されない。
- (9) <body> ~ </body> に本文を記入する。コメント文は、<!-- と --> の間に書く。

2 やってみよう

sample0.html のコピー sample1.html を作り、<title> と <body> の中身を次のように書いてみよう。

sample1.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>HTML 入門</title>
<script>
  // ここには、JavaScript 言語で記述します。
</script>
</head>
<body>

<font color="red">                                <!--赤色を指定 -->
<h1 align="center">はじめての HTML</h1> <!-- 見出しには、<h1> ～ <h6> を用いる。-->
</font>
<font color="purple" align="left">
<h2>草枕</h2>
</font>
<font color="blue">
<h3>夏目漱石 </h3>
</font>
<font color="green">
  <p> 山路を登りながら、こう考えた。</p>
  <p> 智に働けば角が立つ。情に棹させば流される。意地を通せば窮屈だ。
  とかくに人の世は住みにくい。</p>
  <p> 住みにくさが高じると、安い所へ引き越したくなる。どこへ越しても住みにくいと悟った時、
  詩が生れて、画が出来る。</p>
</font>

</body>
</html>
```

結果は、sample1.html で確認できます。

注

- (1) 文字の色指定は、``タグで行うのが便利である。指定方法は、red(赤), orange(橙), yellow(黄), green(緑), blue(青), indigo(藍), purple(紫), black(黒), gray(灰色), white(白) などとカラー名を書くか、R(赤)、G(緑)、B(青) の各色を 1 バイト=8 ビット、すなわち、 $2^8 = 256$ 段階で表す。例えば、"red" は、"#ff0000" のように表す。
- (2) 文書の位置は、align(整列の意味) で指定します。左寄せは left, 中寄せは center、右寄せは right です。
- (3) 改行は `
` ですが、段落 (paragraph) を表す場合には、その段落を `<p>` と `</p>` で囲みます。

3 もっとやってみよう

3.1 フォントの種類

今まで使ったフォントは標準のものでした。フォント (font) の種類を指定するには、font タグに face = " …" を追加します。ここで、…にフォント名を書きます。それらには、

fantasy	: 装飾フォント
cursive	: 草書体系フォント
HG 行書体	: 行書体系フォント
serif	: 明朝系フォント
sans-serif	: ゴシック系フォント
monospace	: 等幅フォント

などがあります。ただし、PC やブラウザによって見え方が違うようです。

3.2 色の種類

HTML で様々な色を使ってみよう。

- (1) 色は英語または、数字で指定します。例えば、赤は red, または ff0000 です。色の英語名を覚えましょう。
- (2) 背景色の指定は、bgcolor="色" で指定します。bg は、background(背景) の略です。
- (3) ` ~ ` は、番号なしの箇条書きに用います。ul は、unordered list(順序付けられていないリスト) の略です。`<ls>` の後にリストの項目を書きます。
- (4) いわゆる色の違いを色相と言います。黒っぽいか白っぽい、色の明るさを明度と言います。色には、もう一つ、色の艶 (つや) やかさを表す彩度があります。この 3 つを合わせて、色の三属性と言います。

sample1-1.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>HTML 色入門</title>
<script>
</script>
</head>

<body bgcolor="skyblue">

<font color="red">
<h1>HTML の色見本</h1>

<h2>色相</h2>
</font>
<ul>
  <li> <font color="red">赤 (red)</font>
  <li> <font color="orange">橙 (orange)</font>
  <li> <font color="yellow">黄 (yellow)</font>
  <li> <font color="green">緑 (green)</font>
  <li> <font color="blue">青 (blue)</font>
  <li> <font color="indigo">藍 (indigo)</font>
  <li> <font color="purple">紫 (purple)</font>
</ul>

<h2>明度</h2>
<ul>
  <li> <font color="black">黒 (black)</font>
  <li> <font color="gray">灰色 (gray)</font>
  <li> <font color="darkgray">暗灰色 (darkgray)</font> <!-- あんかいしょく -->
  <li> <font color="lightgray">ライトグレー (lightgray)</font>
  <li> <font color="white"><span style="background-color:black">白 (white)</span></font>
</ul>

</body>
```

結果は、sample1-1.html で確認できます。

4 css 入門

HTML のスタイルは、HTML とは別の所に記述するのが良いとされています。そのために使うのが CSS で、これは Cascading Style Sheet の略です。

CSS の書き方

css の書き方は以下の通りです。

```
セクタ {  
    プロパティ: 値;  
    プロパティ: 値;  
    ⋮  
}
```

ここで、セクタには、タグ名を書きます。このタグでかこまれた要素の位置や、フォントのサイズ、色などの性質 (プロパティ) を値で指定します。この値とは、数値の場合が多いですが、必ずしも数値とは限りません。例えば、位置のプロパティを指定する場合には、absolute や relative などが用いられます。また、フォントの色を指定する場合の値は red のように英語の色名を用いることもできます。

4.1 漱石先生の写真

それでは、sample1.html を css を用いて書き直しましょう。ついでに、漱石先生の写真も貼り付けてみましょう。画像を挿入するときには、

```

```

などと書きます。ここで、img は、image(画像) の略です。イメージファイルとしては、JPG, GIF などを用います。alt は、altnet(代役) の略で、画像そのものが表示できないときに、文字の『画像』を表示します。

以上の修正をおこなった後のソースファイルは、次のようになります。

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>HTML 入門</title>
<script>
</script>

<style>
body{
    position:relative;
}
h1{
    color : red;
    text-align : center;
}
h2{
    color: purple;
    text-align: left;
}
h3{
    color: blue;
}
div{
    color: green;
}
div2{
    position:absolute;
    left: 850px;
    top: 50px;
}
</style>
</head>
<body>

<h1>はじめての HTML</h1>

<h2>草枕</h2>

<h3>夏目漱石</h3>
```

```
<div>
  <p> 山路を登りながら、こう考えた。</p>
  <p> 智に働けば角が立つ。情に棹させば流される。意地を通せば窮屈だ。
  とかくに人の世は住みにくい。</p>
  <p> 住みにくさが高じると、安い所へ引き越したくなる。どこへ越しても住みにくいと悟った時、
  詩が生れて、画が出来る。</p>
</div>

<div2>

</div2>

</body>
</html>
```

どうですか。<body> … </body> の中は、すっきりしました。結果は、sample1-2.html で確認できます。

4.2 壁紙を張る

壁紙を張る場合には、css で貼り方を指定します。張り紙の指定は、body で行います。

```
sample1-3.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>壁紙</title>
<style>
body{
    background-size: cover; // 背景を画像で覆う。
}
</style>
<script>
    // ここには、JavaScript 言語で記述します。
</script>
</head>
<body background="back1_4.jpg">
    <!--
        壁紙 (back1_4.jpg) の指定
    -->
<center>
<h1>壁紙を張る</h1>
</center>

</body>
</html>
```

結果は、sample1-3.html で確認できます。

5 ハイパーリンクの作成と youtube 動画の埋め込み

文書中のある部分から、別のページにジャンプできるようにすることをリンクを張ると言います。その書き方は、以下の通りです。

```
<a href="ジャンプ先の URL">リンクを張る文字列</a>
```

例えば、文書中に「基礎力の青森大学」という言葉があって、そこから青森大学のホームページのトップに移動できるようにするためには、


```
<a href="http://www.aomori-u.ac.jp/">基礎力の青森大学</a>
```

のように書きます。

最近のホームページでは、YouTube の動画を埋め込んだサイトを良く見かけます。これを行うには、以下のようにします。最も簡単な方法は、

1. 埋め込みたい YouTube の動画に移動する。
2. 動画を右クリックすると、メニューが現れる。
3. 「埋め込みコードをコピー」を左クリックする。
4. 作成中のウェブサイトのコピーしたコード **Crtl** を押しながら **C** キーをクリックして貼り付ける。

もう一つの方法は、

1. 埋め込みたい YouTube の動画に移動する。
2. 動画の下にある下図のような [共有] ボタンをクリックする。

3. [埋め込みコード] をクリックする。
4. ボックスに表示されている **HTML** コードをコピーする。
5. 作成中のウェブサイトのコピーしたコードを貼り付ける。

それでは、sample2.html を作ってみましょう。ソースコードは以下の通りです。

sample2.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title></title>
<script>
    // ここには、JavaScript 言語で記述します。
</script>
</head>
<body>

<center>
<font face="fantasy">
<h1>
The fundamental things apply as time goes by.
</h1>
</font>
<font face="HG 行書体">
<h2>
基本的な事柄は、時の過ぎゆく中で通用する。
</h2>
</font>
<a href="http://www.aomori-u.ac.jp/">
<font face="serif"><h1>基礎力の青森大学</h1></font></a>
</font>
<br>
<br>
    <iframe width="336" height="252" src="https://www.youtube.com/embed/EPSzkpNucV8?rel=0&autoplay=1"
        frameborder="0" allowfullscreen></iframe>
</center>

</body>
</html>
```

結果は、sample2.html で確認できます。

ここで、埋め込みコードを貼り付けた後に、少し手を加えました。コードを貼り付けた時、

```
src="https://www.youtube.com/embed/EPSzkpNucV8"
```

となっていたところに、?rel=0&autoplay=1 を加え、

```
src="https://www.youtube.com/embed/EPSzkpNucV8?rel=0&autoplay=1"
```

としました。`?rel=0`を加えることにより、動画の再生が終わったあとに、他の動画の関連づけをおこないません。`&autoplay=1`でウェブページを立ち上げたときに、ビデオが自動的に再生されます。

6 フォーム機能

フォーム機能とは、テキスト入力、ボタン、チェックボックス、ラジオボタンなど、ユーザーからの入力を受け付ける様式 (form) のことです。これにより、送信側と受信側でデータの送受信が可能となります。

フォームの要素をウェブに付け加えるには、`<input>` タグを用います。例えば、テキストとボタンを付け加えるには、以下のようになります。

sample3-1.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title></title>
<script>
  // ここには、JavaScript 言語で記述します。
</script>
</head>
<body>

<form name="f1">
  <input type="text" size="30">
  <br>                                <!-- 改行 -->
  <input type="button" name="btn" value="ボタン">
</form>

</body>
</html>
```

結果は、sample3-1.html で確認できます。

テキストフィールドとボタンは付けることができましたが、これだけでは何も実行することができません。フォームを操作する命令は、JavaScript で行います。

7 JavaScript 入門 (1)

sample3-1.html では、web ページにホーム機能を付け加えました。これだけでは、何の面白み也没有せん。それでは、テキストに入力した値を取得したり、ボタンを押すとページを更新したりするような機能を持たせてみましょう。そのためには、JavaScript を用います。これは、Web ページ動作を制御する目的でつくられたスクリプト言語です。

最も簡単な例として、テキストフィールドに名前を入力して、ボタンを押すと「こんにちは、〇〇さん。」と名前を返すプログラム (sample3-2.html) を作ってみましょう。

sample3-2.html

```
<html>
<head>
<title>ご挨拶</title>
<script>

function onClick(){
    var str1=document.getElementById("txt").value;
    var str2=document.getElementById("display");
    str2.innerHTML=' こんにちは'+str1+' さん';
}

</script>
</head>
<body>
<form name="f1">
    <input type="text" size="30" id="txt">
    <br>
    <input type="button" name="button" value="ボタン" onclick="onClick()">
    <br>
    <p id="display"></p>
</form>

</body>
</html>
```

プログラムの説明

- (1) ボタンの後にパラグラフ `<p id="display"></p>` を加えます。
- (2) テキストフィールドとパラグラフに、各々、txt, display と名前 (id) を付けます。
`<input type="text" size="30" id="txt">`
`<p id="display"></p>`
- (3) ボタンに onclick 属性を記述して、ボタンを押したときに JavaScript で書かれた関数 onClick() を呼び出します。
`<input type="button" name="button" value="ボタン" onclick="onClick()">`

(4) テキストフィールドに書かれた内容 (value) を変数 str1 に代入します。

```
var str1=document.getElementById("txt").value;
```

(5) パラグラフ <p id="display"></p> の内容を変数 str2 に代入します。

```
var str2=document.getElementById("display");
```

もちろん、パラグラフには、何も書かれていませんから、str2 の中身は空っぽです。

(6) str2 の内容を、'こんにちは'+str1+'さん' と書き換えることは、次のようにして行います。

```
str2.innerHTML='こんにちは'+str1+'さん';
```

結果は、sample3-2.html で確認できます。

8 JavaScript 入門 (2) 更なる改良

sample3-2.html で、当初の目的は達成されましたが、テキストフィールドが空欄のままの場合に、ボタンを押すと、「こんにちは」とおかしいことになります。そこで、テキストが空っぽの場合は、名前の入力进行を促すように改良しましょう。そのためには、script に、以下命令を付け加えるだけです。

```
if(str1)
    str2.innerHTML=' こんにちは'+str1+' さん';
else
    str2.innerHTML = ' 名前を入力してください。';
```

sample3-3.html

```
<html>
<head>
<title>ご挨拶 (2)</title>
<script>

function onClick(){
    var str1=document.getElementById("txt").value;
    var str2=document.getElementById("display");
    if(str1)
        str2.innerHTML=' こんにちは'+str1+' さん';
    else
        str2.innerHTML = ' 名前を入力してください。';

}
</script>
</head>
<body>
<form name="f1">
    <input type="text" size="30" id="txt">
    <br>
    <input type="button" name="button" value="ボタン" onclick="onClick()">
    <br>
    <p id="display"></p>
</form>

</body>
</html>
```

結果は、sample3-3.html で確認できます。

9 応用例題

それでは、今まで学習したことをもとに、「質問コーナー」(sample3f.html)を作成してみましょう。

```
<html>
<head>
<title>質問コーナー</title>
<style>
div1{
    position:relative;
    left: 300px
    top: 0px;
}
</style>
<script>
    document.write('<br><br><br>');
function onClick(){
    var str1=document.getElementById("txt").value;
    var str2=document.getElementById("display");
    if(str1){
        str2.innerHTML='<h3>友よ、答は風の中に。<br><br>' +
        '<font face="cursive">The answer, my friend, is blowing in the wind. <br>' +
        'The answer is blowing in the wind. </font></h3>'+
        '<center><iframe width="420" height="315" ' +
        'src="https://www.youtube.com/embed/vWwgrjjIMXA?rel=0" ' +
        ' frameborder="0" allowfullscreen></iframe></center>';
    }else{
        str2.innerHTML = '<font color="red"><b> 質問を入力してください。</b></font>';
    }
}
</script>
</head>
<body>
<div1>
<center>
<h2><font face="serif"> 質問コーナーです。何でもお気軽にお尋ねください。</font></h2>
<br>
<form name="f1">
<textarea type="text" rows="5" cols="60" id="txt" ></textarea>
<br>
```



```
<br>
<input type="button" name="button" value="回答" onclick="onClick()">
<br>
<p id="display"></p>
</form>
</center>
</div1>
</body>
</html>
```

結果は、sample3f.html をご覧ください。

縦書きに挑戦

次に、縦書きに挑戦します。題材は、歌舞伎『三人吉三廓初買』の「大川端庚申塚の場」における有名な長ゼリフです。「そんなのは知らねえ」と言う人もいるでしょうが、決めのセリフ、

「こいつぁ春から 縁起がいいわえ」

は、聞いたことがある人も多いのではないのでしょうか。

```
<!doctype html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
body{
  padding: 20px;
  margin-left: -200px;
  writing-mode: tb-rl; //IE
  writing-mode: vertical-rl;
  font-family: HGP 行書体;
}

h1{
  font-size: 30px;
}

h2{
  font-size: 26px;
  color: black;
```

```

}

div{
  color: darkblue;
  font-size: 24px;
}

div2{
  position: relative;
  left: 830px;
  top: 270px;
}

</style>
<script>
</script>
</head>
<body>
<br>
<br>
<h1><ruby>三人吉三廓初買<rt>さんになきちさくるわのはつがい</h1>
<h2>
      <ruby>河竹黙阿弥<rt>かわたけもくあみ</rt></ruby></h2>
<br>
<div>
<p><ruby>月<rt>つき</rt></ruby>も<ruby>朧<rt>おぼろ</rt></ruby>に
<ruby>白魚<rt>しらうお</rt></ruby>の </p>
<p><ruby>篝<rt>かがり</rt></ruby>も<ruby>霞<rt>かす</rt></ruby>む 春の空 </p>
<p>冷てえ風も ほろ<ruby>酔<rt>よい</rt></ruby>に </p>
<p><ruby>心持<rt>こころも</rt></ruby>ちよく うかうかと </p>
<p>浮かれ<ruby>鳥<rt>がらす</rt></ruby>の <ruby>只一羽<rt>ただいちわ</rt></ruby> </p>
<p><ruby>疳<rt>ねぐら</rt></ruby>へ帰る <ruby>川端<rt>かわばた</rt></ruby>で </p>
<p><ruby>棹<rt>さお</rt></ruby>の<ruby>雫<rt>しずく</rt></ruby>か
<ruby>濡手<rt>ぬれて</rt></ruby>で<ruby>粟<rt>あわ</rt></ruby> </p>
<p>思いがなく 手に<ruby>入<rt>い</rt></ruby>る百両 </p>

<p>ほんに今夜は <ruby>節分<rt>せつぶん</rt></ruby>か </p>
<p>西の海より 川の中 </p>
<p>落ちた<ruby>夜鷹<rt>よたか</rt></ruby>は <ruby>厄落<rt>やくおと</rt></ruby>し </p>
<p><ruby>豆沢山<rt>まめだくさん</rt></ruby>に <ruby>一文<rt>いちもん</rt></ruby>の </p>
<p><ruby>銭<rt>ぜに</rt></ruby>と違って <ruby>金包<rt>かねづつ</rt></ruby>み </p>
<p>こいつあ春から <ruby>縁起<rt>えんぎ</rt></ruby>がいいわえ </p>

```

```
<br>
</div>

<div2 class="box">

</div2>

</body>
</html>
```

結果は、sample4.html でご覧いただけます。

10 ビデオの公開

ビデオを公開する方法としては、YouTube に代表される動画配信サービスを利用する方法が一般的です。サービスにアップロードした動画を 3.2 節で説明した方法でウェブページに貼り付けることができます。ここでは、動画配信サービスを利用しないで済みます方法について解説します。

10.1 リンクを張る

最も手っ取り早い方法は、ウェブページにビデオのハイパーリンクを作成する方法です。例えば、同じフォルダに置かれた AodaiPV-ver01.mp4 にリンクを張る場合には、

```
<A href="AodaiPV-ver01.mp4"> 青森大学 1 </A>
```

と書くだけです。以下に例を示します。

sample5-1.html

```
<!doctype html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
body{
  padding: 50px;
  margin: 30px;
}
</style>
<script>
</script>
</head>
<body>
<center>
  <h1>ビデオによる青森大学紹介</h1>
  <h3>これは 2014 年に制作したものです。</h3>
  <A href="AodaiPV-ver01.mp4"> 青森大学 1 </A><br>
  <br>
  <A href="AodaiPV-ver02.1.mp4" >青森大学 2 </A>
</center>
</body>
</html>
```

結果は、sample5-1.html でご覧いただけます。

10.2 動画を埋め込む

動画の埋め込みには、<iframe> タグを利用します。これは、3.2 で YouTube の動画を埋め込むときに使ったものです。iframe は、inline frame の略で、ある枠組み (frame) の中に別の枠組み (frame) を作ることです。次の例では、

```
<iframe src="AodaiPV-ver01.mp4" width="690" height="419"></iframe>
```

の部分で埋め込みを行っています。

sample5-2.html

```
<!doctype html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
body{
  padding: 50px;
  margin: 30px;
}
</style>
<script>
</script>
</head>
<body>

<center>
  <h1>ビデオによる大学紹介</h1><br>
  <h3>これは 2014 年に制作したものです。</h3>
  <iframe src="AodaiPV-ver01.mp4" width="690" height="419"></iframe>
</center>

</body>
</html>
```

結果は、sample5-2.html でご覧いただけます。

10.3 複数の動画の再生

今回のデモに用いた動画は、2 部構成です。これを 9.2 節の方法で 2 つの inline frame を用いて行くと、間が抜けているように見えます。そこで、JavaScript を使って 1 つのフレームで 2 つの動画を切り替えて再生するようにします。この切り替えは、onClick() 関数を用いて行います。動画の切り替えと同時に、ボタンの表示も変更します。

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>タイトル </title>
<style>
body{
    padding: 50px;
    margin : 30px;
    top: 0px;
}
</style>
<script>
    var str1;
function onClick(){
    var image=document.getElementById('image');
    var btn = document.getElementById('btn');
    if(str1){
        image.src='AodaiPV-ver01.mp4';
        btn.innerHTML="第 2 部へ";
        str1="";
    }else{
        image.src='AodaiPV-ver02.1.mp4';
        btn.innerHTML="第 1 部へ";
        str1="1";
    }
}
</script>
</head>
<body>
<center>
    <h1>ビデオによる大学紹介</h1><br>
    <h3>これは 2014 年に制作したものです。</h3>
    <p> <iframe id="image" width="600" height="390" src="AodaiPV-ver01.mp4"
        frameborder="0" allowfullscreen></iframe> </p>
</center>
    <center>
    <button id="btn" onclick="onClick()">第 2 部へ</button><br>
    </center>

```

```
<br>  
  
</body>  
</html>
```

結果は、sample5-3.html でご覧いただけます。

10.4 複数動画の再生 2

ここでは、任意の数の動画をボタンで切り替える方法を示します。以下の例では、4つの動画を扱います。YouTube上の4つのSukiyakiの動画をボタンで切り替えて表示し、聴き比べができるようにします。選択したボタンが分かるように、文字を赤色に変えます。

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
</style>
<script>
onload= function(){
    var b=document.getElementsByTagName('button');

    str0 = "https://www.youtube.com/embed/C35DrtPlUbc?rel=0";
    str1 = "https://www.youtube.com/embed/AMQ0AdqkXGs?rel=0";
    str2 = "https://www.youtube.com/embed/pSYS7cPc3nE?rel=0";
    str3 = "https://www.youtube.com/embed/v0zewZucBXI?rel=0";
    str4 = "https://www.youtube.com/embed/WeCAQmxq_Ew?rel=0";

    image.src=str0+"&autoplay=0" ;

    b[0].style.color='red';

    b[0].onclick = function(){
        for(var i=0; i<5; i++)
            b[i].style.color='black';
        b[0].style.color='red';
        image.src=str0 ;
    }
    b[1].onclick = function(){
        for(var i=0; i<5; i++)
            b[i].style.color='black';
        b[1].style.color='red';
        image.src= str1 ;
    }
    b[2].onclick = function(){
        for(var i=0; i<5; i++)
```



```

        b[i].style.color='black';
    b[2].style.color='red';
    image.src= str2;
}
b[3].onclick = function(){
    for(var i=0; i<5; i++)
        b[i].style.color='black';
    b[3].style.color='red';
    image.src= str3;
}
b[4].onclick = function(){
    for(var i=0; i<5; i++)
        b[i].style.color='black';
    b[4].style.color='red';
    image.src= str4;
}
};

</script>
</head>
<body>
<center>
<h1>Sukiyaki 聴き比べ</h1>
<button id="btn0" > Kyu Sakamoto </button>
<button id="btn1" >Hikaru Nishida</button>
<button id="btn2" >    4P.M.    </button>
<button id="btn3" >A Taste of Honey</button>
<button id="btn4" >Kenny Ball and His Jazzmen</button>
<p>
<iframe id="image" width="600" height="390"
</iframe> </p>
</center>
</body>
</html>

```

結果は、sample5-4.html でご覧いただけます。

10.5 YouTube ビデオの Loop 再生

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="shift_jis">
<title> </title>
<script>

</script>
</head>
<body>
<center>
<h1>
YouTube ビデオの Loop 再生
</h1>
<br>
<h2>
The Circle Game
</h2>
<iframe width="420" height="315"
src="https://www.youtube.com/embed/ybyAgZtWLh8?rel=0&loop=1&playlist=ybyAgZtWLh8"
frameborder="0" allowfullscreen>
</iframe>
</body>
</html>
```

結果は、sample5-5.html でご覧いただけます。

11 サウンド

11.1 コントローラによる再生

```
sound1.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> </title>
<style>
  body{
    padding: 30px;
    margin-left: 100px;
    font-size : 20px;
  }
</style>
</head>
<body>
<h1><p> サウンドテスト</p></h1>
<h2><p> コントローラの表示</p></h2>
440 ヘルツの音
<br>
<audio src="sin.wav" controls></audio>
<br>
<br>
440 ヘルツと 442 ヘルツの音の唸り
<br>
<audio src="beat.wav" controls></audio>
<br>
<br>
音色 (440 ヘルツと倍音の重ね合わせ)
<br>
<audio src="timbre.wav" controls></audio>
</body>
</html>
```

結果は、sound1.html でご覧いただけます。

11.2 自動再生 (オートプレイ)

```
saound2.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title>sound</title>
<style>
  body{
    padding: 30px;
    margin-left: 100px;
    font-size : 20px;
  }
</style>
<script>
</script>
</head>
<body>
<h1>
<p> サウンドテスト (オートプレイ)</p>
</h1>
<br>
sin2.wav の再生
<br>
<audio src="sin2.wav" autoplay>
</audio>
```

結果は、sound2.html でご覧いただけます。

11.3 ループ再生

sound3.html

```
<!DOCTYPE html>
<html lang="ja">

<head>
<meta charset="utf-8">
<title>sound</title>
<style>
  body{
    padding: 30px;
    margin-left: 100px;
    font-size : 18px;
  }
</style>
</head>

<body>
<h2>
<p> サウンドテスト (ループ再生)</p>
</h2>
<br>
siren.wav の再生
<br>
<audio source src="siren.wav" autoplay loop>
</audio>
</body>

</html>
```

結果は、sound3.html でご覧いただけます。

11.4 JavaScript による再生

```
sound4.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> sound</title>
<style>
  body{
    padding: 30px;
    margin-left: 200px;
    font-size : 60px;
    color : red;
  }
</style>
<script>
onload=function(){
  var aud = document.getElementById("aaa");
  aud.play();
};
</script>

</head>
<body>
<p> JavaScript によるサウンドの再生 1</p>
<p>注意! 画面を開くと警告音が鳴ります。</p>
<br>
<br>
<audio id="aaa" src="beep.wav"></audio>
</body>
</html>
```

結果は、sound4.html でご覧いただけます。

11.5 異なるサウンドの連続再生

```
sound5.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> sound</title>

<script>
var aud = document.getElementsByTagName("audio");
onload=function(){
    aud[0].play();
    var dt = aud[0].duration;
    var n = (dt+0.5)*1000;
    setTimeout('aud[1].play()', n);
};
</script>
</head>

<body>
<h1>
<p>JavaScript によるサウンドの再生 2</p>
</h1>
<h2>
<p>異なる 2 つのサウンドを続けて再生</p>
</h2>
<br>
<audio id="aaa" src="doremi.wav"></audio>
<audio id="abc" src="doshira.wav"></audio>
</body>
</html>
```

結果は、sound5.html でご覧いただけます。

11.6 異なる2つのサウンドを続けて有限回再生

```
sound6.html
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> sound</title>

<script>
var aud = document.getElementsByTagName("audio");
onload=function(){
    var dt0 = aud[0].duration;
    var dt1 = aud[1].duration;
    var n0 = (dt0+0.1)*1000;
    var n1 = (dt1+0.1)*1000;

    for(var i=0; i<10; i++){
        setTimeout('aud[0].play()', i*n0+i*n1);
        setTimeout('aud[1].play()', (i+1)*n0+i*n1);
    }
};
</script>
</head>
<body>
<h1>
<p>JavaScript によるサウンドの再生 3</p>
</h1>
<h2>
<p>異なる 2 つのサウンドを続けて有限回再生</p>
</h2>
<br>
<audio id="aaa" src="doremi.wav"></audio>
<audio id="abc" src="doshira.wav"></audio>
</body>
</html>
```

結果は、sound6.html でご覧いただけます。

11.7 異なる2つのサウンドを続けて loop 再生

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> sound</title>
<script>
var aud = document.getElementsByTagName("audio");
    var dt0;
    var dt1;
    var n0;
    var n1;

    function test(){
        setTimeout('aud[1].play()', (i+1)*n0+i*n1);
        i++;
    };

onload=function(){
    dt0 = aud[0].duration;
    dt1 = aud[1].duration;
    n0 = (dt0+0.1)*1000;
    n1 = (dt1+0.1)*1000;

    dd = Math.min(n0, n1);
    ddd = dd/100;

    i=0;
    aud[0].play();
    setInterval('test()', ddd);
    setInterval('aud[0].play()', n0+n1);
};
</script>
</head>
<body>
<h1>
<p>JavaScript によるサウンドの再生 4</p>
</h1>
<h2>
```

```
<p>異なる 2 つのサウンドを続けて loop 再生</p>
</h2>
<br>
<audio id="aaa" src="doremi.wav"></audio>
<audio id="abc" src="doshiraS.wav"></audio>
</audio>
</body>
</html>
```

結果は、sound7.html でご覧いただけます。

12 YouTube 動画を OFF してオーディオプレーヤーとして使用する

12.1 YouTube をオーディオプレーヤーとして使用 1

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title> sound</title>
<script>
onload= function(){
    var sl=document.getElementById('sel1');
    var im=document.getElementById('image');

    sl.onchange = function(){
        var selectedItem = this.options[this.selectedIndex];
        if(selectedItem.value == 0){
            im.width='600';
            im.height='390';
        }else if(selectedItem.value == 1){
            im.width='0';
            im.height='0';
        }
    }
}

var b=document.getElementById('btn');
b.onclick = function(){
    image.src="https://www.youtube.com/embed/0XSaKQlBZuE?rel=0&autoplay=1" ;
}
};
</script>
</head>
<body>
<center>
<h1>
<p>YouTube 動画を OFF してオーディオプレーヤーとして使用 1 </p>
</h1>
<br>
動画 (ON/OFF)
<select id="sel1">
```

```
<option value="0">ON</option>
<option value="1" >OFF</option>
</select>
<br>
<br>
<button id="btn" >Rock'n'Roll Music <br> Chuck Berry </button>
<br>
<p>
<iframe id="image" width="640" height="360"
src="https://www.youtube.com/embed/0XSaKQ1BZuE?rel=0"
frameborder="0" allowfullscreen></iframe>
</p>
</center>
</body>
</html>
```

結果は、sound8.html でご覧いただけます。

12.2 YouTube をオーディオプレーヤーとして使用 2

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
</style>
<script>
onload= function(){
    var sl=document.getElementById('sel1');
    var im=document.getElementById('image');

    str0 = "https://www.youtube.com/embed/w40ushYAaYA?rel=0";
    str1 = "https://www.youtube.com/embed/J2bm6KQKHeQ?rel=0&autoplay=1" ;
    str2 = "https://www.youtube.com/embed/OT1HCQcSHW0?rel=0&autoplay=1" ;
    str3 = "https://www.youtube.com/embed/JnuB_DjhCOM?rel=0&autoplay=1" ;
    str4 = "https://www.youtube.com/embed/arPSIL-LamA?rel=0&autoplay=1" ;

    image.src=str0+"&autoplay=0" ;

    sl.onchange = function(){
        var selectedItem = this.options[this.selectedIndex];
        if(selectedItem.value == 0){
            im.width='600';
            im.height='390';
        }else if(selectedItem.value == 1){
            im.width='0';
            im.height='0';
        }
        console.log( selectedItem.value );
    }

    var b=document.getElementsByTagName('button');
    b[0].style.color='red';

    b[0].onclick = function(){
        for(var i=0; i<5; i++)
            b[i].style.color='black';
    }
}
```

```

    b[0].style.color='red';
    image.src=str0+"&autoplay=1" ;
}
b[1].onclick = function(){
    for(var i=0; i<5; i++)
        b[i].style.color='black';
    b[1].style.color='red';
    image.src=str1;
}
b[2].onclick = function(){
    for(var i=0; i<5; i++)
        b[i].style.color='black';
    b[2].style.color='red';
    image.src= str2;
}
b[3].onclick = function(){
    for(var i=0; i<5; i++)
        b[i].style.color='black';
    b[3].style.color='red';
    image.src= str3;
}
b[4].onclick = function(){
    for(var i=0; i<4; i++)
        b[i].style.color='black';
    b[4].style.color='red';
    image.src= str4;
}
};
</script>
</head>
<body>
<center>
<h1>
<p>YouTube 動画を OFF してオーディオプレーヤーとして使用 2</p>
</h1>
<h2>
雨の曲 5 つ
</h2>

動画 (ON/OFF)
<select id="sel1">

```

```
<option value="0">ON</option>
<option value="1" >OFF</option>
</select>
<br>
<br>
<button id="btn0" >Singin' In The Rain <br> Gene Kelly </button>
<button id="btn1" >Rhythm Of The Rain <br> Cascades </button>
<button id="btn2" >Rain Drops Keep Falling On My Head <br> B.J. Thomas </button>
<button id="btn3" >Have You Ever Seen The Rain? <br> John Fogerty </button>
<button id="btn4" >L'a Pioggia <br> Gigliola Cinquetti</button>
<br>
<br>
<p>
<iframe id="image" width="600" height="390"
</iframe> </p>
</center>
</body>
</html>
```

結果は、sound9.html でご覧いただけます。

12.3 YouTube をオーディオプレーヤーとして使用 3

ここでは、JavaScript, YouTube api(application programming interface), jQuery を用います。更に、HTML と JavaScript のファイルを分けます。

HTML ファイル (sound10.html)

```
sound10.html
<!DOCTYPE html>
<html>
<script src="sample3.js"></script>
<script src="jquery-3.2.1.min.js"></script>
<script>
</script>
<body>
<BR>
<center>
<h1>
<p>オーディオプレーヤーとしての YouTube 3</p>
</h1>
<h2>
Jazz の調べ
</h2>
<select id="sel1">
  <option value='0'>Take Five</option>
  <option value='1' >Take The A Train</option>
  <option value='2' >In The Mood</option>
</select>
<br>
<br>
<button id="play">再生</button>
<button id="pause">一時停止</button>
<button id="stop">早戻し</button>
<br>
  <div id="sample1"></div>
  <div id="sample2"></div>
  <div id="sample3"></div>
<br>
</center>
</body>
</html>
```

結果は、sound10.html でご覧いただけます。

js ファイル (sample3.js)

```
// IFrame Player API の読み込み
var tag = document.createElement('script');
tag.src = "https://www.youtube.com/iframe_api";
var firstScriptTag = document.getElementsByTagName('script')[0];
firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

function onYouTubeIframeAPIReady() {
    ytPlayer1 = new YT.Player(
        'sample1', // 埋め込む場所
        {
            width: '0', // プレーヤー幅
            height: '0', // プレーヤー高さ
            videoId: 'Tn3acz2KI9M'
        }
    );
    ytPlayer2 = new YT.Player(
        'sample2',
        {
            width: '0',
            height: '0',
            videoId: 'KV8Hj_E8LJc'
        }
    );
    ytPlayer3 = new YT.Player(
        'sample3', // 埋め込む場所の指定
        {
            width: '0',
            height: '0',
            videoId: 'omjZ73GFPaQ'
        }
    );
};

onload = function(){
    var sl=document.getElementById('sel1');

    $('#play').click(function() {
        ytPlayer1.playVideo();
    });
};
```

```

});
$('#pause').click(function() {
    ytPlayer1.pauseVideo();
});
$('#stop').click(function() {
    ytPlayer1.pauseVideo().seekTo(0);
});

sl.onchange = function(){

    var selectedItem = this.options[this.selectedIndex];

    if(selectedItem.value == 0){
        $('#play').click(function() {
            ytPlayer2.pauseVideo();
            ytPlayer3.pauseVideo();
            ytPlayer1.playVideo();
        });
        $('#pause').click(function() {
            ytPlayer1.pauseVideo();
        });
        $('#stop').click(function() {
            ytPlayer1.pauseVideo().seekTo(0);
        });
    }else if(selectedItem.value == 1){
        $('#play').click(function() {
            ytPlayer1.pauseVideo();
            ytPlayer3.pauseVideo();
            ytPlayer2.playVideo();
        });
        $('#pause').click(function() {
            ytPlayer2.pauseVideo();
        });
        $('#stop').click(function() {
            ytPlayer2.pauseVideo().seekTo(0);
        });
    }else if(selectedItem.value == 2){
        $('#play').click(function() {
            ytPlayer1.pauseVideo();
            ytPlayer2.pauseVideo();
            ytPlayer3.playVideo();
        });
    }

```

```
});  
$('#pause').click(function() {  
    ytPlayer3.pauseVideo();  
});  
$('#stop').click(function() {  
    ytPlayer3.pauseVideo().seekTo(0);  
});  
}  
}  
};
```

13 HTML と JavaScript ファイルを分ける

13.1 プルダウンメニュー、option, switch

HTML ファイル (sample6.html)

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> </title>
<style>
</style>
<script src="sample6.js" >    <!-- JavaScript ファイル (sample6.js) を読み込みます -->
</script>
</head>
<body>
<center>
<h1>
<p>「確率・統計学」講義</p>
</h1>
<h2>
残念ながら、私の講義ではありません。
</h2>
<select id="sel1">
  <option value= '0' >Stat 110 Harvard - Let's Get Started (Statistics Song)</option>
  <option value= '1' >Lecture 1: Probability and Counting </option>
  <option value= '2' >Lecture 2: Story Proofs, Axioms of Probability </option>
  <option value= '3' >Lecture 3: Birthday Problem, Properties of Probability </option>
  <option value= '4' >Lecture 4: Conditional Probability</option>
  <option value= '5' >Lecture 5: Conditioning Continued, Law of Total Probability </option>
  <option value= '6' >Lecture 6: Monty Hall, Simpson's Paradox </option>
  <option value= '7' >Lecture 7: Gambler's Ruin and Random Variables </option>
  <option value= '8' >Lecture 8: Random Variables and Their Distributions </option>
  <option value= '9' >Lecture 9: Expectation, Indicator Random Variables, Linearity </option>
  <option value= '10' >Lecture 10: Expectation Continued </option>
  <option value= '11' >Lecture 11: The Poisson distribution </option>
  <option value= '12' >Lecture 12: Discrete vs. Continuous, the Uniform </option>
  <option value= '13' >Lecture 13: Normal distribution </option>
  <option value= '14' >Lecture 14: Location, Scale, and LOTUS </option>
  <option value= '15' >Lecture 15: Midterm Review </option>
```

```

<option value= '16' >Lecture 16: Exponential Distribution </option>
<option value= '17' >Lecture 17: Moment Generating Functions </option>
<option value= '18' >Lecture 18: MGFs Continued </option>
<option value= '19' >Lecture 19: Joint, Conditional, and Marginal Distributions </option>
<option value= '20' >Lecture 20: Multinomial and Cauchy </option>
<option value= '21' >Lecture 21: Covariance and Correlation </option>
<option value= '22' >Lecture 22: Transformations and Convolutions </option>
<option value= '23' >Lecture 23: Beta distribution </option>
<option value= '24' >Lecture 24: Gamma distribution and Poisson process </option>
<option value= '25' >Lecture 25: Order Statistics and Conditional Expectation </option>
<option value= '26' >Lecture 26: Conditional Expectation Continued </option>
<option value= '27' >Lecture 27: Conditional Expectation given an R.V.</option>
<option value= '28' >Lecture 28: Inequalities </option>>
<option value= '29' >Lecture 29: Law of Large Numbers and Central Limit Theorem </option>
<option value= '30' >Lecture 30: Chi-Square, Student-t, Multivariate Normal </option>
<option value= '31' >Lecture 31: Markov Chains</option>
<option value= '32' >Lecture 32: Markov Chains Continued </option>
<option value= '33' >Lecture 33: Markov Chains Continued Further </option>
<option value= '34' >Lecture 34: A Look Ahead </option>
<option value= '35' >Joseph Blitzstein: "The Soul of Statistics" </option>
</select>
<br>
<p>
<iframe id="image" width="600" height="390"
src="https://www.youtube.com/embed/ulStCvohAHk?rel=0"
frameborder="0" allowfullscreen></iframe> </p>
</center>
</body>
</html>

```

結果は、sample6.html でご覧いただけます。

js ファイル (sample6.js)

```
onload= function(){
    var sl=document.getElementById('sel1');
    var image=document.getElementById('image');

    sl.onchange = function(){
        var selectedItem = this.options[this.selectedIndex];

        switch (selectedItem.value){
            case '0': image.src="https://www.youtube.com/embed/ulStCvohAHk?rel=0" ; break;
            case '1': image.src="https://www.youtube.com/embed/KbB0FjPg0mw?rel=0" ; break;
            case '2': image.src="https://www.youtube.com/embed/FJd_1H3rZGg?rel=0" ; break;
            case '3': image.src="https://www.youtube.com/embed/LZ5Wergp_PA?rel=0" ; break;
            case '4': image.src="https://www.youtube.com/embed/P7NE4WF8j-Q?rel=0" ; break;
            case '5': image.src="https://www.youtube.com/embed/JzDvVgNDxo8?rel=0" ; break;
            case '6': image.src="https://www.youtube.com/embed/fDcjhAKuhqQ?rel=0" ; break;
            case '7': image.src="https://www.youtube.com/embed/PNrQcDslGi4?rel=0" ; break;
            case '8': image.src="https://www.youtube.com/embed/k2BB0p8byGA?rel=0" ; break;
            case '9': image.src="https://www.youtube.com/embed/LX2q356N2rU?rel=0" ; break;
            case '10': image.src="https://www.youtube.com/embed/P1fSFvhPf7Q?rel=0" ; break;
            case '11': image.src="https://www.youtube.com/embed/TD1N4hxxMzY?rel=0" ; break;
            case '12': image.src="https://www.youtube.com/embed/Tci---bVs60?rel=0" ; break;
            case '13': image.src="https://www.youtube.com/embed/72QjzHnYvL0?rel=0" ; break;
            case '14': image.src="https://www.youtube.com/embed/9vp1L12NpRw?rel=0" ; break;
            case '15': image.src="https://www.youtube.com/embed/yFRZf81sB5k?rel=0" ; break;
            case '16': image.src="https://www.youtube.com/embed/bM6nFDjvEns?rel=0" ; break;
            case '17': image.src="https://www.youtube.com/embed/N806zd6vTZ8?rel=0" ; break;
            case '18': image.src="https://www.youtube.com/embed/tVDdx6xU0cs?rel=0" ; break;
            case '19': image.src="https://www.youtube.com/embed/J70dP_AECzQ?rel=0" ; break;
            case '20': image.src="https://www.youtube.com/embed/xiVWNkQUqKk?rel=0" ; break;
            case '21': image.src="https://www.youtube.com/embed/IujCYxtpszU?rel=0" ; break;
            case '22': image.src="https://www.youtube.com/embed/yXwPUAIVfyg?rel=0" ; break;
            case '23': image.src="https://www.youtube.com/embed/UZjlBQbV1KU?rel=0" ; break;
            case '24': image.src="https://www.youtube.com/embed/Qjeswpm0cWY?rel=0" ; break;
            case '25': image.src="https://www.youtube.com/embed/2LR5JYbhyjg?rel=0" ; break;
            case '26': image.src="https://www.youtube.com/embed/PgawcWisb0I?rel=0" ; break;
            case '27': image.src="https://www.youtube.com/embed/gjBvCiRt8QA?rel=0" ; break;
            case '28': image.src="https://www.youtube.com/embed/UtXK_EQ3Pow?rel=0" ; break;
            case '29': image.src="https://www.youtube.com/embed/OprNqnHsVIA?rel=0" ; break;
            case '30': image.src="https://www.youtube.com/embed/MF-XSJ0sGqw?rel=0" ; break;
```

```
case '31':image.src="https://www.youtube.com/embed/8AJPs3gvNlY?rel=0" ; break;
case '32':image.src="https://www.youtube.com/embed/aBG0yZv2pZE?rel=0" ; break;
case '33':image.src="https://www.youtube.com/embed/Q-pCzTpWPBU?rel=0" ; break;
case '34':image.src="https://www.youtube.com/embed/ChS3K20-h7o?rel=0" ; break;
case '35':image.src="https://www.youtube.com/embed/dzFf3r1yph8?rel=0" ; break;
}
}
};
```

14 YouTube 動画の playlist による連続再生

sample7.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title></title>
<script>
  // ここには、JavaScript 言語で記述します。
</script>
</head>
<body>

<center>
<h1>
<p>YouTube 動画の playlist による連続再生</p>
</h1>
<h2>
Waterloo Road,  Les Champs-Elysees,  オー・シャンゼリゼ
</h2>
<br>

<iframe width="769" height="432"
src="https://www.youtube.com/embed?playlist=aKYUrxWRLHk,oamRCeLNAWA,rPieHg98SKM&autoplay=1&loop=0"
frameborder="0" allowfullscreen></iframe>

</center>

</body>
</html>
```

結果は、sample7.html でご覧いただけます。

15 YouTube 動画の部分再生 1

sample8.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title> sample8 </title>
<script>
onload= function(){
    var b=document.getElementsByTagName('button');
    b[0].style.color='red';
    b[0].onclick = function(){
        for(var i=0; i<2; i++)
            b[i].style.color='black';
        b[0].style.color='red';
        image.src="https://www.youtube.com/embed/6Gj5aLQIEQo?rel=0&autoplay=1" ;
    }
    b[1].onclick = function(){
        for(var i=0; i<2; i++)
            b[i].style.color='black';
        b[1].style.color='red';
        image.src="https://www.youtube.com/embed/6Gj5aLQIEQo?start=225&end=230&rel=0&autoplay=1" ;
    }
};
</script>
</head>
<body>
<center>
<h1>YouTube 動画の部分再生</h1>
<h2> Gone with the Wind - End Scene </h2>
<button id="btn0" > 全て再生 </button>
<button id="btn1" >一部再生</button>
<p>
<iframe id="image" width="600" height="390"
src="https://www.youtube.com/embed/6Gj5aLQIEQo?rel=0"
frameborder="0" allowfullscreen></iframe> </p>
</center>
</body>
</html>
```

結果は、sample8.html でご覧いただけます。

16 YouTube 動画の部分再生 2

sample9.html

```
<html>
<head>
<title>sample9</title>
<script>

  function onClick(){
    image.src="https://www.youtube.com/embed/EbgAu_X2mm4?start=237&end=242rel=0&autoplay=1";
  }
</script>
</head>

<body>

<center>
<h1>これも YouTube 動画の部分再生</h1>
<A href="#" onclick="onClick()">
<h2>
Wir m&uuml;ssen wissen, wir werden wissen.
<h2></A>
<p>
<h3>
我々は知らねばならない。我々は知るであろう。
<p>
David Hilbert(ダフィット・ヒルベルト)
</h3>
<iframe id="image" width="0" height="0" src="" frameborder="0" allowfullscreen>
</iframe>
<p> <p>
<br>
【注】文章をクリックすると、(少し時間がかかりますが)Hilbert 先生の読み上げる声を聞くことができます。
</center>

</body>
</html>
```

結果は、sample9.html でご覧いただけます。

17 YouTube を BGM として

sample10.html

```
<html>
<head>
<title>sample10</title>
<script></script>
</head>

<body>

<center>
<h1>YouTube を BGM として</h1>
<BR>
<h3>
Moon River, Charade, and My Fair Lady Overture
</h3>
<center>

<iframe width="1" height="1"
src="https://www.youtube.com/embed/28f9bqklFeo?autoplay=1&playlist=NZ1ANlvn3rs,PpVse2tNaSE&loop=1"
frameborder="0" allowfullscreen>
</iframe>

</body>
</html>
```

結果は、sample10.html でご覧いただけます。