



スクールテキストリーダー (School Text Reader)

■スクールテキストリーダーについて

スクールテキストリーダー (School Text Reader) は、簡単にデジタル副読本を作成することができるツールです。WEB サーバーにアップロードすることでインターネット環境と JavaScript が動作するブラウザがあればどこからでも閲覧することができるようになります。

■主な機能

- ・画面の縦横の向きにあわせて、画面いっぱいに表示することができます。
- ・ペンツール機能で書き込みができます。※書き込んだ内容は保存できませんが、画像としてダウンロードが可能です。
- ・リンク機能を設定することで指定のページへジャンプしたり、外部サイトを開いたりすることができます。
- ・右綴じ・左綴じどちらにも対応しています。

※ページデータは画像で用意する必要があります。読み込み可能なデータはブラウザで扱える画像フォーマットです。(jpeg や PNG、WEBP 等。)

■実装手順

1. サイトからファイル (schooltextreader***.zip) をダウンロードします。
2. 任意のフォルダに展開すると main と sample の2つのフォルダが作成されます。

※sample フォルダはサンプルデータです。必要なければ削除して構いません。
3. main フォルダの中の pic フォルダの中にページデータを入れます。
4. main フォルダの中の setting フォルダの中に「setting.js」ファイルをメモ帳などで開き、設定を行います。※設定方法の詳細については後述で説明します。
5. main フォルダをサーバーにアップロードして完了です。

※main フォルダの名前は任意に変更して構いません。

■利用にあたって

本コンテンツは副読本の作成を目的としたツールです。学校教育以外の目的での利用や、著作権に違反するページデータを利用することはおやめください。

■免責事項

「スクールテキストリーダー(School Text Reader)」の利用者は、自己の責任において本コンテンツを利用するものとし、利用により発生した全ての結果について一切の責任とリスクを負うものとします。コンテンツを利用した事で何らかの被害および損害が発生したとしても、コンテンツ著作者は一切の責任を負いません。また、コンテンツ著作者はバージョンアップ、不具合修正の義務を負いません。

■setting.js の設定について

各項目について、以下に説明していきます。

```
//タイトルを設定。文字列なので「"」（ダブルクォーテーション）で囲むこと  
  
let book_title = "タイトル";  
  
let book_sub_title = "サブタイトル";  
  
let book_copy = "著作者";
```

読み込み時に表示されるタイトル、サブタイトル、著作者を設定します。

文字列なのでダブルクォーテーション囲んでください。各項目の中を書き換えるだけです。

```
//ファイル名が連番の前に文字列がある場合に設定 test_10.png の場合は"test_"  
  
let filename = "sample";
```

ページデータのファイル名を設定します。ファイル名の最後は必ずページ番号になっている必要があります。文字列+ページ番号の場合に、その文字列を設定します。もし番号だけなら文字列は不要なので「let filename = ""」と空値にしてください。

```
//ファイル番号の0埋め。  
  
//true:[xx009,xx010,xx011...] false:[xx9,xx10,xx11...]
```

```
//埋める桁数は下の総ページ設定数から算出
```

```
let fileNo_rept = false;
```

1 ページ目のファイル名が「001」と最大ページの桁に合わせて 0 がついている場合は「true」そうでなければ「false」を指定します。文字列ではないのでダブルクォーテーションで囲む必要はありません。

```
//ページ画像の拡張子"png","jpeg","webp"など（大文字小文字注意）
```

```
let filetype = "PNG";
```

ページデータの拡張子を設定します。大文字、小文字が区別される場合がありますので、正確に設定してください。なお、ページデータはすべて同じ拡張子である必要があります。

```
//総ページ数
```

```
let page_num = 2;
```

総ページ数を指定します。ページデータのファイル数から自動で算出はされません。

```
//開始ページ数
```

```
let start_page = 1;
```

開いた時に最初に表示するページを指定します。標準は最初の表紙ページの 1 になっています。

```
//右綴じ:true 左綴じ:false  
  
let Right_binding = false;
```

本の右綴じ・左綴じを true と false で設定します。

```
//本の1ページの用紙(画像)サイズを[幅,高さ](px)で指定。  
  
const page_size = [723,1024];
```

ページデータの画像サイズをピクセル単位で指定します。横と縦をカンマで区切って設定します。

```
//見開きのサイズと縦型のサイズの[幅,高さ]  
  
//上のページサイズが入るように設定。余白を含めて切りが良い数値に設定する。  
  
//高さは4つ下の項目のUIアイコンのサイズも含めるように設定すること。  
  
let main_canvas_size = [1500,1100]; //見開き用に横2倍のサイズが入る値を設定。  
  
let sub_canvas_size = [750,1100]; //メインキャンバスの半分の値を設定。
```

アプリの解像度を設定します。main_canvas_size は見開き用のサイズでページサイズの2倍が収まる値にすると良いです。高さは後述のUIの高さを含めた値より大きい値を設定してください。例はページサイズ 723×2=1446 なので 1500px 高さ 1024+50=1074 なので 1100px としています。sub_canvas_size の縦画面はUIサイズが 70px と横画面よりも大きく設定していますが、1024+70 で 1094 なので、同じ 1100px に収まっています。

※以降の設定項目は必要に応じて任意に行ってください。

```
//初期の表示 0:1 ページ 1:見開き (2 ページ)
```

```
let show_mode = 0;
```

初期表示の設定です。1 にすると開始時から見開きモードになります。

```
//同一ウィンドウで外部へのリンク移動時の閲覧状況保存
```

```
//0:保存無し 1:現在のページとビューを保存 2:現在ページ及び拡大率や表示位置な  
ども保持
```

```
let save_setting = 2;
```

外部リンクで同じウィンドウで他のサイトに移り、再び戻ってきたときに表紙に戻るのではなく、直前に見ていたページを表示するのか、また拡大表示などの

```
//設定データの保持期間 (分)
```

```
let save_period = 60;
```

閲覧状態の保存を 1 または 2 にした場合、その保存期間を分単位で指定します。他のサイトへ遷移してから、この時間を超えて戻った場合は表紙に戻されます。

```
//背景色 color1 は横向き画面時、color2 は縦向き画面時
```

```
const backgroud_color1= "#e6e6e6";
```

```
const backgroud_color2= "#e6e6e6";
```

背景色のカラーコードを指定します。画面が縦の時と横の時で別の色を指定できます。

```
//ページに影をつける
```

```
let add_shadow = true;
```

ページに影を付けて立体的に見せる場合は true、しない場合は false を設定します。

```
//影の色
```

```
let shadow_color = "gray";
```

影の色を指定します。例では gray とカラーネームで設定していますが、#ffffff の様なカラーコードでも問題ありません。

```
//影のぼかし、値が大きいほどぼける。
```

```
let shadow_Blur = 8;
```

ページの影のぼかし具合を指定します。値が大きいほど、影がぼやけて広がります。

```
//UI アイコン一つ分のサイズ V:縦型時 H:横型時
```

```
let UI_size_V = 50;
```

```
let UI_size_H = 70;
```

下部の UI の縦のサイズを指定します。縦画面と横画面を切り替えた際、同じサイズで指定していると、解像度の関係でサイズが異なって表示されることがあるので、調整が必要です。デフォルトは縦画面では 50px、横画面では 70px としています。

```
//UI のベースカラー  
  
const backgroud_colorUI= "#e6e6e6";
```

UI のベースカラーを指定します。

```
//ローディングアイコン&文字色  
  
const loading_color = "#000000";
```

データ読み込み中のアイコンと文字の色を指定します。

```
//ダークモードを true にするとアイコンが白になる  
  
//フルスクリーン時は必ずダークモード  
  
let dark_mode = false;
```

ダークモードを true にするとアイコンが白いデザインになり、背景が黒になります。false にしていても、フルスクリーンボタンを押すと強制的にダークモードになります。

```
//本棚（ホームボタン）の設定  
  
let home_btn = true;  
  
let home_url = "https://~ ";//空値なら一つ上の階層へ
```

home_btn を true にすると UI に本棚アイコン（ホームボタン）を表示します。また home_url に URL を入れると特定のページにリンクし、空値「''」なら、1 つ上のフォルダ階層にリンクします。


```
//描画ツールボタンの設置&サイズ (px)
```

```
let draw_btn = true;
```

```
let draw_btn_size_V = 50;
```

```
let draw_btn_size_H = 70;
```

描画モードで表示されるツールアイコンのサイズを指定します。これも UI のサイズを同様に縦画面と横画面で表示されるサイズが変わるので、調整が必要です。基本は、UI のサイズと同様の値で問題ありません。

```
//ペンと消しゴムのサイズリスト
```

```
let pen_size_list = [1,3,7,12,18,25];
```

```
let marker_size_list = [10,20,30,40,50,60];
```

```
let eraser_size_list = [5,10,15,25,40,65];
```

ペンとマーカー、消しゴムで描画するサイズを指定します。サイズは6段階あり、それぞれどれくらいのサイズかを個別に指定できます。ページ画像のサイズに合わせて調整してください。※値は配列になっているので、カンマで区切ってください。

```
//最大拡大率
```

```
let max_expand = 3.0;
```

UI の虫眼鏡ツールやピンチ操作で拡大できる最大拡大率を設定します。

```
//拡大アイコンの拡大率（ボタンを押すごとにどれくらい変化するか）
```

```
let expand_scale = 0.5;
```

拡大縮小アイコンを押したときの拡大率の変化量を設定します。アイコンを押した際はここで指定した値の倍数に強制的に設定されます。

```
//スライダーの高さ
```

```
let slider_height = 30;
```

ページ送りのスライダーの高さを設定します。

```
//スライダーつまみの最低幅
```

```
//スライダーのつまみはスライダー全体のページ数割だが、
```

```
//ページ数が多いとつまみが小さくなりすぎるので下限値を設定
```

```
let slider_width = 100;
```

スライダーのつまみアイコンに最低幅を設定します。

```
//スライダーのページ数表示の値ずらし
```

```
//表紙を含めた時など画像に記載されたページ番号と合わない時に設定
```

```
let difference_page = 0;
```

スライダーに表示されるページ番号の値をずらします。例として表紙等があり、実質ページデータの3つ目から1ページが始まる場合は-2を設定。

```
//ページめくり判定の幅の割合  
  
//0.2 の場合、ページ幅の 20%の左右をタッチすると、ページめくりを行う  
  
let page_side_per = 0.2;
```

ページ画像の左右をタップした際に、ページめくりが実行されるサイズの設定です。ページサイズに対しての割合で設定してください。

```
//スワイプページめくりを実行する判定時間（ミリ秒）と判定距離及び角度  
  
let swipe_time = 500;  
  
let swipe_dist = 50;  
  
let swipe_angle = 25;
```

スワイプしてページめくりを行う際の設定です。例では 500 ミリ秒以内に 50 ドット以上のスワイプ操作を行い、かつスワイプ開始地点からスワイプ終了地点の角度が斜め 25 度以内に収まっている時にページめくりが実行されます。

```
//ページ拡大時は左右タッチでページめくりを行わない  
  
let change_page_block_expanded = false;
```

true を設定した場合、ページを拡大している時は、タッチ操作でページめくりを禁止します。

```
//ページをめくるアニメーションを行う

let change_page_animation = true;

//1ページめくるアニメーション時間(ミリ秒)

let change_page_time = 250;
```

ページをめくる時のアニメーションと、めくるアニメーションの時間を設定します。

```
//ページロード失敗時のリロード試行回数&待機時間(ミリ秒)&リロードボタンカラー

let reload_num = 4;

let reload_time = 3000;

let reload_btn_color = "#fff";
```

読み込み失敗時にリロード試行回数と試行までの待機時間、ボタンの色を指定します。

リロードボタンの色は前述にあったローディングアイコン&文字色の loading_color と同じにならないようにしてください。

```
/ページリンクの設定

//ページ番号,開始 x 座標(px),開始 y 座標(px),幅(px),高さ(px),リンク先 0:ページ
1:URL(同ウィンドウ) 2: URL (別ウィンドウ) ,ページ番号 or リンク先 URL

//ページ番号はスライダのページ差分ではなくファイル順の番号なので注意すること

let link_color = "#00f";
```

```
let page_link = new Array();

function set_pagelink(){

page_link = [

    [0,0,0,0,0,0,0],

    [0,0,0,0,0,0,0]//最後はカンマ不要

];

}
```

ページのリンクを設定します。link_color はページ内のリンク個所にマウスカーソルを合わせた時に表示されるエリアの色を指定します。

リンク情報は常に更新される仕組みにしているので、ブラウザの開発ツール等を使うと、実際のページを見ながらリアルタイムに数値を調整できるようにしています。

コメントの説明にもあるように、[ページ番号,x 座標,y 座標,横幅,高さ,リンクの種別,リンク先]の配列をつなげていきます。

リンクの種別は 0 がページ移動、1 がリンク先へ移動、2 は別ウィンドウを開いてリンク先へ移動となっています。0 を指定してページ移動にした場合、ページ先は数値（ページ数）で指定しますが、これは最初を 1 ページとした場合のページ数です。表示などを差し引く設定をした時に画面上に表示されるページ数ではないので注意が必要です。