目次

[**1.概要** 2](#_Toc162783440)

[**2.htmlの説明** 2](#_Toc162783441)

[**3.cssの説明** 2](#_Toc162783442)

[.parent{} 2](#_Toc162783443)

[.res{} 2](#_Toc162783444)

[**4.変数の説明** 2](#_Toc162783445)

[const変数 2](#_Toc162783446)

[let変数 2](#_Toc162783447)

[**5.使用しているメソッドやプロパティの説明** 3](#_Toc162783448)

[**6.処理の流れ** 3](#_Toc162783449)

[① 変数の宣言 3](#_Toc162783450)

[② クリック時の処理 3](#_Toc162783451)

[**7.コメントなしのプログラム（bodyタグ以下）** 4](#_Toc162783452)

ランダム単語表示ツール

# **1.概要**

このツールはセットされた単語をランダムで表示するものである。

一度表示された単語は一周するまでは出てこないようになっている。

# **2.htmlの説明**

<div class=res>:<div class=res>を画面中央にするための親クラス

<div class=res>:単語を表示するためのタグ

# **3.cssの説明**

## .parent{}

中央揃え:flexを使用して上下中央揃えをしている。高さを指定すること。

## .res{}

フォントサイズは70pxで、user-selectはnoneにしてできないようにする。(クリックしてドラッグ表示されるのが煩わしいため。)

# **4.変数の説明**

## const変数

|  |  |
| --- | --- |
| result | 単語を表示する< div class=res>の要素を取得する |
| MAXNUM | str配列の長さを入れておく変数。これをもとにランダムの幅を決める |

## let変数

|  |  |
| --- | --- |
| str[] | 任意の単語を入れるための配列 |
| r | ランダムな値を入れておくための変数 |
| hold[] | rに入れた値ですでに出ている値を記憶しておくための配列 |
| flg | whileでループを抜けるスイッチの役割を持つ。  flg=0はhold配列のなかにrと同じものがあった状態（既出の単語）を表し、ループを続ける。flg=1の場合は逆に配列の中に同じものがなかった状態（新出の単語）を表し、ループを抜けられる。 |

# **5.使用しているメソッドやプロパティの説明**

Math.floor():()の中にある数値を、小数点以下で切り捨て

参考: [Math.floor() - JavaScript | MDN (mozilla.org)](https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/Math/floor)

hold.push():配列の末尾に()の要素を追加する

[Array.prototype.push() - JavaScript | MDN (mozilla.org)](https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/Array/push)

hold.length:配列の個数を調べる

[String: length - JavaScript | MDN (mozilla.org)](https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/String/length)

# **6.処理の流れ**

1. 変数の宣言

htmlのdiv(class=res)タグを取得。

strに単語を[配列]で入れる。（let宣言）

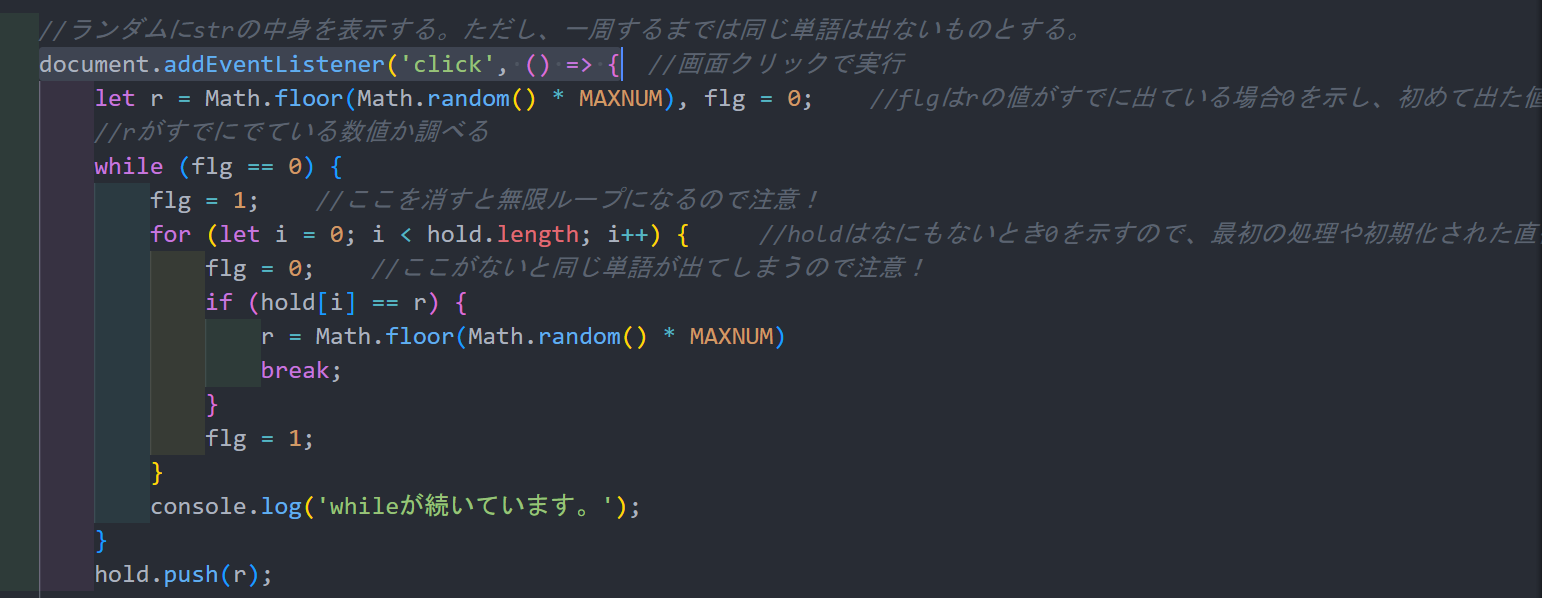
MAXNUMにstrの長さを入れる。(const宣言)

hold配列を宣言し、初期化する（let宣言）

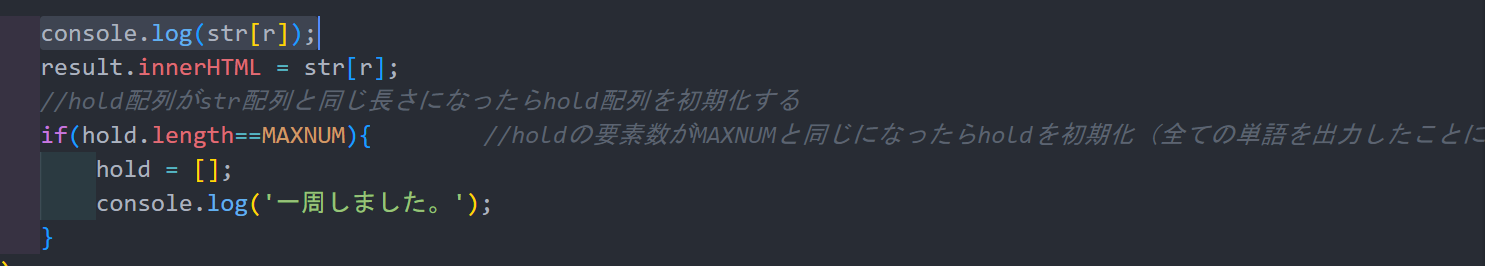
1. クリック時の処理
2. 画面クリックで実行する。
3. rを宣言し、ランダムの数値を入れる。ただし、上限は単語の長さになる(10個の場合、0~9までの数値になる)
4. flgを宣言し0を代入する。
5. rがすでに出た値か調べる。

whileの中

1. flgに1を代入する（これをしないとholdは最初や初期化した直後は0を示すため、下のfor文が実行されず、flgが一生0のままで無限ループになる。）
2. for文をhold配列の長さ分繰り返す。（最初や初期化した直後はholdの長さが0と示すためこのfor文は実行されない）
3. flgに0を代入する。
4. holdの中にrと同じものがあればその時点でrに新たなランダム数を代入し、breakしてfor文を抜ける。( if(hold[i] == r){} )
5. （4）が実行されなければflgに1を代入し、hold配列の最後まで1が代入されたらfor文を終え、while文を抜ける。
6. while文を抜けたら、現在のrの値をhold配列の末尾に追加する。（hold.push()）



1. 結果を画面に表示するため、resを取得したresultをinnerHTMLで加工する。
2. もしhold配列がMAXNUMと同じならholdを初期化する。



# **7.コメントなしのプログラム（bodyタグ以下）**

