

JavaScriptのおさらい

■特記事項■

「HTML/CSSのおさらい」
の作成データが必要になるので
そちらがまだできていない方は
完成させたくうえで実施してください

「HTML/CSSのおさらい」で作成したフォルダを
コピー&ペーストなどで複製し
フォルダ名を「JavaScript_rev」としたうえで
以降は「JavaScript_rev」で作業してください

作業工程

- 【1】 JavaScriptファイルの読み込み
- 【2】 配列の作成・使用
- 【3】 繰り返し処理
- 【4】 プリロード機能の実装
- 【5】 時間の取得・使用
- 【6】 クッキーの使用
- 【7】 オブジェクトの作成・使用
- 【8】 チャレンジ課題

JavaScriptファイルの読み込み

【1】

各ページに最新版のjQueryのファイルを適用できるように、
作成済みのHTMLファイルの内容を変更

▼ヒント

- ・ jQueryのファイルは、教本p247を参考に各自ダウンロード

※ダウンロードするjQueryのファイルは、
最新版のデータ量が圧縮された本番用プログラム

配列の作成・使用

【2-1】

「image.html」や「put_cards.html」などで使用している
「トランプ画像」のファイル名に注目して、
「マーク」の配列と「数字」の配列を作成

※ページへの反映は【2-2】で実施します

【条件】

- ・ジョーカーと裏面のカード画像は除く

▼ヒント

この配列の情報は後の処理においてプログラム中で読み込むため、
画像ファイルのファイル名に使われている文字列のうち

- ・すべてのカードに共通している文字列
- ・マークを示している文字列
- ・数字を示している文字列

に着目し、共通点・相違点を探して考えてみましょう

配列の作成・使用

【2-2】

「image.html」にて、
定義した配列を用いて、任意のトランプ画像をページ上に表示

▼ヒント

- ・「任意の親要素に子要素を追加する方法」
- ・「取得した要素のコンテンツをHTMLで書き換えるjQueryのメソッド」

など、HTMLに表示する方法はたくさんありますが、
自分がやりやすい形でOKです

※いろいろな実現方法があるため答えは一つではないです

▼画像を表示するために考慮する必要がある内容のヒント

- ・画像を表示するタグ
- ・ファイルが格納されているフォルダまでのパス
- ・ファイル名のうち「共通部分」となる文字列
- ・ファイル名のうち「マーク」を示す文字列
- ・ファイル名のうち「数字」を示す文字列
- ・ファイル名のうち「拡張子」を示す文字列

配列の作成・使用

【2-3】

「image.html」にて、
ページを読み込むたびに、表示画像がランダムに変化するように変更

▼ヒント

- ・ランダムに何かをする処理が必要になります
- ・配列の値をランダムで指定することになるので、
値をどう指定するかがポイントです

繰り返し処理

【3】

「table.html」にて、
「繰り返し処理」を用いて九九表を作成

▼ヒント

- ・ 繰り返し処理の理解と、どんな順番で処理するかが重要です
- ・ 表は「行」と「列」で構成されていることを踏まえて考えてみましょう

プリロード機能の実装

【4】

「image.html」 および 「put_cards.html」 において、
imageフォルダに用意されているトランプ画像を
あらかじめすべて読み込むプリロード機能を実装

※デベロッパーツールのネットワークタブの情報を参照すると、
表示している画像以外の画像も読み込めているかどうかを確認できます

▼ヒント

JavaScriptの教本p241を参考に実装してみましょう

- ・ 【2-1】 で作成した配列を使用する
- ・ 【2-2】 で使用した画像ファイル名を生成する機能を流用する
- ・ 【3】で行った繰り返し処理を応用し、
ファイル名を繰り返し生成する処理を実装する
- ・ 配列では処理できない画像も含まれているので要注意

時間情報の取得と利用

【5】

「index.html」に対して、index.htmlを表示した際に、時間帯に応じて変化する挨拶をダイアログボックスで表示する機能を実装しましょう

※ほかのページでは挨拶が表示されないようにすること

【おまけ】

HTMLの<noscript>タグを使って、JavaScript非対応端末に向けたメッセージをindex.htmlに設定

※教本に載っていないタグですが、各自で調べて実装してみましょう

Cookieの使用

【6】

「index.html」に対して、ページの配色などを変更し、その設定がCookieに保存されているような仕組みを実装しましょう

▼条件

- ・ 変更した配色は、index.html以外のページでも有効になるようにすること
- ・ 一度ブラウザを閉じるなどした後、再度ページにアクセスすると、変更の設定が有効になっている状態にする

※Cookieを使用するので、動作確認には「Served」を利用する必要があります

オブジェクトの作成・使用

【7】

「put_cards.html」において、
表示されるポーカーの役がページを読み込むたびに
ランダムで変わるようにする

▼盛り込む要素

- ・表示される役は、ページを読み込むたびにランダムで変わるようにする
- ・ポーカーの役は各自インターネットで検索し、
「役名」「使用するカード」「説明文」を含む
オブジェクト形式のデータとして用意する

チャレンジ課題

ここまでに作成しているHTMLファイルをもとに、

「challenge_01.html」「challenge_02.html」「challenge_03.html」
を作成して、チャレンジ1～3の内容を盛り込んだページを作成してみましょう

チャレンジ課題の内容については
仕組みを自分なりに考えてコーディングしてもらえるとベストですが、
インターネットで検索すると作成方法が出てくるので、写経してもOKです

書き写した後は、コードの内容を自分なりに読み解いてみて
読み解いた内容が後で見直せるように、コメントで入れておきましょう

チャレンジ課題

【チャレンジ1】

ページ上で操作できる計算機を作成しましょう

▼考え方

- ・市販の計算機を参考に、
計算機にはどんな機能が実装されているのか考えてみましょう

【チャレンジ2】

入力した任意の10進数の数を、
16進数・2進数に変換して表示するプログラムを作成しましょう

※余力があれば、入力した任意の16進数・2進数などの数を、
10進数に変換して表示するプログラムも作成してみましょう

【チャレンジ3】

ユーザーが任意の数を入力すると、
その数までの素数をページ上に表示する機能

※ 素数を求めるプログラムは処理が複雑なので検索推奨 ※