

app5.jsのドキュメント

hello1とhello2について

ファイル一覧

| ファイル名 | 説明 |
|----------------|--------------------------|
| app5.js | プログラム本体 |
| views/show.ejs | hello1とhello2のテンプレートファイル |

hello1のプログラムの内容

`const message1 = "Hello world"; const message2 = "Bon jour";` Hello worldとBon jourの二つの文字列をそれぞれ変数`message1`,`message2`に格納する.

`res.render('show', { greet1:message1, greet2:message2});` `show.ejs`に`message1`,`message2`を渡す

hello2のプログラムの内容

`res.render('show', { greet1:"Hello world", greet2:"Bon jour"});` 定義された文字列を`show.ejs`に渡す.

使用方法

1. `app5.js`を起動する
2. Webブラウザで`localhost:8080/hello1` or `hello2`にアクセスする.

iconについて

ファイル一覧

| ファイル名 | 説明 |
|-----------------------------|-----------------|
| app5.js | プログラム本体 |
| views/icon.ejs | iconのテンプレートファイル |
| public/Apple_logo_black.svg | 表示する画像 |

プログラムの内容

`res.render('icon', { filename:"./public/Apple_logo_black.svg", alt:"Apple Logo"});` `icon.ejs`に画像ファイルを渡す.

使用方法

1. `app5.js`を起動する

2. Webブラウザでlocalhost:8080/iconにアクセスする.

luckについて

ファイル一覧

| ファイル名 | 説明 |
|----------------|-----------------|
| app5.js | プログラム本体 |
| views/luck.ejs | luckのテンプレートファイル |

プログラムの内容

```
const num = Math.floor( Math.random() * 6 + 1 );
let luck = '';
if( num==1 ) luck = '大吉';
else if( num==2 ) luck = '中吉';
else if( num==3 ) luck = '吉';
else if( num==4 ) luck = '末吉';
else if( num==5 ) luck = '凶';
else if( num==6 ) luck = '大凶';
```

Math.floorで1から6のランダムな数字を生成し,1から6をそれぞれ大吉,中吉,吉,末吉,凶,大凶に割り当てる.

console.log('あなたの運勢は' + luck + 'です');引いた運勢を表示する.

res.render('luck', {number:num, luck:luck});: luck.ejsにnumber(ランダムな数)とluck(運勢)を渡す.

使用方法

- 1. app5.jsを起動する
- 2. Webブラウザでlocalhost:8080/luckにアクセスする.

jankenについて

ファイル一覧

| ファイル名 | 説明 |
|--------------------|-------------------|
| app5.js | プログラム本体 |
| views/janken.ejs | jankenのテンプレートファイル |
| public/janken.html | じゃんけん開始画面 |

プログラムの内容

```
let hand = req.query.hand;
let win = Number(req.query.win);
let total = Number(req.query.total);
```

プレイヤーの手,勝利数,試合数を格納する.

`const num = Math.floor(Math.random() * 3 + 1);` 1から3のランダムな数字を生成

```
let cpu = '';
if (num === 1) cpu = 'グー';
else if (num === 2) cpu = 'チョキ';
else cpu = 'パー';
```

最初変数cpuには何も格納されていない. numの数字に応じて変数cpuにグー,チョキ,パーという文字列を格納する.

`let judgement = '';` 勝敗結果を格納するための変数judgementを初期化する.

```
if (hand === cpu) {
  judgement = '引き分け';
}
```

プレイヤーとCPUの手が同じ場合、結果を"引き分け" として設定する.

```
else if (
  (hand === 'グー' && cpu === 'チョキ') ||
  (hand === 'チョキ' && cpu === 'パー') ||
  (hand === 'パー' && cpu === 'グー')
) {
  judgement = '勝ち';
  win += 1; // 勝った場合は勝利数を増やす
}
```

・プレイヤーが"グー" かつ CPU が"チョキ" ・プレイヤーが"チョキ" かつ CPU が"パー" ・プレイヤーが"パー" かつ CPU が"グー" 上記のいずれかが満たされれば,プレイヤーの勝利となりjudgementに"勝ち"を入れ,winを1増やす.

```
else {
  judgement = '負け';
}
```

上記のどれにも当てはまらない場合,プレイヤーが負けたと判断しjudgementに"負け"を設定する.

`total += 1;` 試合数を1増やす.勝敗に関係なく試合が行われるたび,実行される.

```
const display = {
  your: hand,
  cpu: cpu,
  judgement: judgement,
  win: win,
  total: total,
};

res.render('janken', display);
```

janken.ejs にプレイヤーの手,cpuの手,勝敗結果,勝利数,試合数を渡す.

使用方法

1. **app5.js**を起動する
2. Webブラウザで**localhost:8080/public/janken.html**にアクセスする
3. 自分の手を入力する

```
flowchart TD;

start["開始"];
box["グー, チョキ, パーのいずれかを入力する"];
number_of_winning["勝利数を増やす"];
number_of_matches["試合数を増やす"];

start --> box
box --> |勝利| number_of_winning
box --> |負け| number_of_matches
box --> |引き分け| number_of_matches
number_of_winning --> number_of_matches
number_of_matches --> |繰り返し| box
```

luckyについて

ファイル一覧

| ファイル名 | 説明 |
|-------------------|-------------------|
| app5.js | プログラム本体 |
| views/lucky.ejs | jankenのテンプレートファイル |
| public/lucky.html | くじ選択画面 |

プログラムの内容

ユーザーが選択したくじを取得して結果を表示 何も選択せず送信した場合は「くじを選択してください」というメッセージを返す

使用方法

1. `app5.js`を起動する
2. Webブラウザで`localhost:8080/public/lucky.html`にアクセスする
3. くじを選ぶ

matchについて

ファイル一覧

| ファイル名 | 説明 |
|-----------------|------------------|
| app5.js | プログラム本体 |
| views/match.ejs | matchのテンプレートファイル |

プログラムの内容

```
if (!yourName || !partnerName) {  
  return res.render("match", { score: null, message: "両方の名前を入力してく  
ださい。" });  
}
```

:yourNameまたはpartnerNameが未入力である場合、「両方の名前を入力してください。」というメッセージをテンプレートに表示する

`const score = Math.floor(Math.random() * 101);`0から100のランダムなスコアを生成する

`const message = `${yourName}さんと${partnerName}さんの相性スコアは ${score}% です!`;`
名前とスコアを組み合わせて文字列を作り、変数messageに格納する

`res.render("match", { score: score, message: message });`:match.ejsにscoreとmessageを渡す

使用方法

1. `app5.js`を起動する
2. Webブラウザで`localhost:8080/match`にアクセスする
3. 名前を入力する