

K

30秒で川柳を書いてTwitterに共有するWebアプリを作りたい。細かい仕様を考えて。



Webアプリを作成するには、まずWebサイトやWebアプリケーションを構築するためのプログラミング言語やフレームワークを選択する必要があります。次に、WebアプリのUI/UXデザインを考え、ユーザーが30秒以内に川柳を書くことができるようになる必要があります。また、Twitterへの共有機能を実装するためには、Twitter APIを使用することができます。



具体的な仕様としては、次のようなものが考えられます。

1. ユーザーはWebアプリにアクセスすることで、30秒のタイマーがスタートします。
2. ユーザーは30秒以内に川柳を入力し、「投稿する」ボタンをクリックすることで投稿できます。
3. 投稿された川柳はWebアプリ上で一覧表示されます。
4. ユーザーは投稿した川柳をTwitterに共有することができます。そのためには、事前にTwitterアカウントでWebアプリにログインする必要があります。
5. ユーザーがTwitterアカウントでログインすると、WebアプリはTwitter APIを使用して、ユーザーのTwitterアカウントに投稿された川柳を共有することができます。

これらの機能を実現するためには、適切なプログラミング言語やフレームワークを選択し、Twitter APIを使用するための認証手順や、タイマー機能、投稿一覧表示、Twitterへの共有機能などを実装する必要があります。

K

Webアプリにアクセスしたときではなくて、スタートボタンが押されたときにタイマーをスタートするように変更して。



はい、Webアプリにアクセスしたときではなく、スタートボタンが押されたときにタイマーをスタートするようにすることも可能です。



具体的には、次のような手順で実装することができます。

1. WebアプリのUI/UXデザインで、タイマーをスタートするスタートボタンを追加します。
2. ユーザーがスタートボタンを押すと、WebアプリのJavaScriptコードが実行され、タイマーがスタートします。
3. JavaScriptコードでは、30秒のカウントダウンを行なながら、ユーザーが川柳を入力することができます。
4. 30秒経過後には、タイマーが停止し、ユーザーが「投稿する」ボタンをクリックすることで投稿することができます。
5. 投稿された川柳はWebアプリ上で一覧表示され、Twitterへの共有も可能です。

このように、スタートボタンが押されたときにタイマーをスタートするように変更することで、ユーザーがより柔軟に川柳を投稿することができるようになります。

K

投稿機能は付かないから、投稿ボタンも投稿された川柳の一覧表示もいらない。代わりに、30秒経過したらプレイ画面は非表示にして、入力された川柳とTwitterへの共有ボタン、「もう一度遊ぶ」ボタンをリザルト画面として表示するようにして。



はい、投稿機能を付けない場合は、投稿ボタンや投稿された川柳の一覧表示は不要です。その代わりに、30秒経過したらプレイ画面を非表示にし、入力された川柳とTwitterへの共有ボタン、「もう一度遊ぶ」ボタンを表示するようにすることができます。



具体的には、次のような手順で実装することができます。

1. WebアプリのUI/UXデザインで、プレイ画面とリザルト画面を作成します。
2. ユーザーがスタートボタンを押すと、WebアプリのJavaScriptコードが実行され、タイマーがスタートします。
3. JavaScriptコードでは、30秒のカウントダウンを行なながら、ユーザーが川柳を入力することができます。
4. 30秒経過後には、タイマーが停止し、プレイ画面が非表示になり、リザルト画面が表示されます。
5. リザルト画面では、入力された川柳が表示され、Twitterへの共有ボタンが表示されます。
6. ユーザーは「もう一度遊ぶ」ボタンをクリックすることで、プレイ画面が再び表示され、新しい川柳を入力することができます。

このようにして実装することで、ユーザーが30秒以内に川柳を入力し、Twitterに共有することができるWebアプリを作成することができます。

K

川柳のお題を設定したい。スタートボタンが押されたら、お題を自動的に選んで表示するようにして。Twitterへ共有するときにもお題と川柳をセットで共有するようにしたい。



具体的には、次のような手順で実装することができます。

1. WebアプリのJavaScriptコードで、お題を設定します。お題は配列に格納し、JavaScriptの`Math.random()`関数を使用してランダムにお題を選択することができます。
2. ユーザーがスタートボタンを押すと、JavaScriptコードが実行され、お題が表示されます。
3. ユーザーは30秒のタイマーを使用して、お題に沿った川柳を入力します。
4. 30秒経過後には、入力された川柳とお題が表示されます。
5. ユーザーはTwitterへの共有ボタンをクリックすることで、Twitter APIを使用して、お題と川柳をセットでTwitterに投稿することができます。

このようにして実装することで、Webアプリにお題を設定し、スタートボタンが押されたときに自動的にお題を選んで表示し、Twitterへの共有時にもお題と川柳をセットで共有することができるようになります。

K これをHTMLとJavaScriptで実装して。まず、HTMLのコードを教えて。△ □

HTMLとJavaScriptでWebアプリを実装する場合、次のようなHTMLのコードが考えられます。

Copy code

```
<!DOCTYPE html>
<html>
<head>
  <title>30秒川柳</title>
</head>
<body>
  <h1>30秒川柳</h1>

  <!-- お題を表示する -->
  <p id="theme"></p>

  <!-- 入力欄を表示する -->
  <textarea id="input"></textarea>

  <!-- タイマーを表示する -->
  <p id="timer"></p>

  <!-- スタートボタンを表示する -->
  <button id="start">スタート</button>

  <!-- Twitterへの共有ボタンを表示する -->
  <button id="share">Twitterへ共有</button>

  <!-- 川柳を表示する -->
  <p id="haiku"></p>

  <!-- 「もう一度遊ぶ」ボタンを表示する -->
  <button id="play-again">もう一度遊ぶ</button>

  <!-- JavaScriptのスクリプトを読み込む -->
  <script src="app.js"></script>
</body>
</html>
```

このHTMLコードでは、お題を表示する`<p>`要素、入力欄を表示する`<textarea>`要素、タイマーを表示する`<p>`要素、スタートボタンを表示する`<button>`要素、Twitterへの共有ボタンを表示する`<button>`要素、川柳を表示する`<p>`要素、「もう一度遊ぶ」ボタンを表示する`<button>`要素が設定されています。

また、JavaScriptのスクリプトを読み込むために、`<script>`要素を使用して、`app.js`というファイルからJavaScriptコードを読み込んでいます。

K JavaScriptのコードを教えて。△ □

```
// お題を格納する配列
const themes = [
  '桜が咲く',
  '雨が降る',
  '風が吹く',
  '夏が来る',
  '冬が来る',
];

// お題を表示する要素を取得する
const themeElement = document.getElementById('theme');

// 入力欄を取得する
const inputElement = document.getElementById('input');

// タイマーを表示する要素を取得する
const timerElement = document.getElementById('timer');

// スタートボタンを取得する
const startButton = document.getElementById('start');

// Twitterへの共有ボタンを取得する
const shareButton = document.getElementById('share');

// 川柳を表示する要素を取得する
const haikuElement = document.getElementById('haiku');

// 「もう一度遊ぶ」ボタンを取得する
const playAgainButton = document.getElementById('play-again');

// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];
  themeElement.textContent = theme;
  timerElement.textContent = '30秒';
  inputElement.value = '';
  shareButton.style.display = 'block';
  playAgainButton.style.display = 'block';
  const intervalId = setInterval(() => {
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;
    if (remainingSeconds === 0) {
      clearInterval(intervalId);
      shareButton.style.display = 'none';
      playAgainButton.style.display = 'none';
    }
  }, 1000);
});

// Twitterへの共有ボタンがクリックされたときの処理
shareButton.addEventListener('click', () => {
  // 入力された川柳を取得する
  const haiku = inputElement.value;

  // Twitterへの投稿用のURLを作成する
  const url = `https://twitter.com/share?text=${encodeURIComponent(haiku)}&hashtags=30秒川柳`;

  // 新しいウィンドウでTwitterの投稿画面を開く
  window.open(url, '_blank');
});
```

◀ コードの途中から続きを教えて。 ▶

HTMLとJavaScriptでWebアプリを実装する場合、次のようなJavaScriptのコードが考えられます。 □ ▷

```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];
  themeElement.textContent = theme;
  timerElement.textContent = '30秒';

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
      clearInterval(intervalId);

      // 入力された川柳を表示する
      haikuElement.textContent = inputElement.value;

      // Twitterへの共有ボタンを表示する
      shareButton.style.display = 'block';

      // 「もう一度遊ぶ」ボタンを表示する
      playAgainButton.style.display = 'block';
    }
  }, 1000);
});

// Twitterへの共有ボタンがクリックされたときの処理
shareButton.addEventListener('click', () => {
  // 入力された川柳を取得する
  const haiku = inputElement.value;

  // Twitterへの投稿用のURLを作成する
  const url = `https://twitter.com/share?text=${encodeURIComponent(haiku)}&hashtags=30秒川柳`;

  // 新しいウィンドウでTwitterの投稿画面を開く
  window.open(url, '_blank');
});
```



コードの途中から続きを教えて。



JavaScriptでWebアプリを実装する場合、次のようなJavaScriptのコードが考えられます。



Copy code

```
// 「もう一度遊ぶ」ボタンがクリックされたときの処理
playAgainButton.addEventListener('click', () => {
  // お題を非表示にする
  themeElement.textContent = '';

  // 入力欄を初期化する
  inputElement.value = '';

  // タイマーを非表示にする
  timerElement.textContent = '';

  // Twitterへの共有ボタンを非表示にする
  shareButton.style.display = 'none';

  // 川柳を非表示にする
  haikuElement.textContent = '';

  // 「もう一度遊ぶ」ボタンを非表示にする
  playAgainButton.style.display = 'none';
});
```

このようにJavaScriptのコードを実装することで、Webアプリのスタートボタンがクリックされた時にお題をランダムに選択し、30秒のタイマーをスタートさせたり、Twitterへの共有ボタンがクリックされたときにTwitterの投稿画面を開いたり、「もう一度遊ぶ」ボタンがクリックされたときに各種要



スタートボタンが押されたら、スタートボタンを非表示にして、「もう一度遊ぶ」ボタンが押されたら、スタートボタンを表示するようにして、変更箇所だけ示して。



スタートボタンが押されたら、スタートボタンを非表示にするように変更するには、次のようにJavaScriptのコードを変更します。



Copy code

```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = theme;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
      clearInterval(intervalId);

      // 入力された川柳を表示する
      haikuElement.textContent = inputElement.value;

      // Twitterへの共有ボタンを表示する
      shareButton.style.display = 'inline-block';

      // 「もう一度遊ぶ」ボタンを表示する
      playAgainButton.style.display = 'inline-block';
    }
  }, 1000);
});
```

また、「もう一度遊ぶ」ボ



続きを教えて。



```
// 「もう一度遊ぶ」ボタンがクリックされたときの処理
playAgainButton.addEventListener('click', () => {
  // スタートボタンを表示する
  startButton.style.display = 'inline-block';

  // お題を非表示にする
  themeElement.textContent = '';

  // 入力欄を初期化する
  inputElement.value = '';

  // タイマーを非表示にする
  timerElement.textContent = '';

  // Twitterへの共有ボタンを非表示にする
  shareButton.style.display = 'none';

  // 川柳を非表示にする
  haikuElement.textContent = '';

  // 「もう一度遊ぶ」ボタンを非表示にする
  playAgainButton.style.display = 'none';
});
```

このようにJavaScriptのコードを変更することで、「もう一度遊ぶ」ボタンが押された時にスタートボタンを表示することができます。

K 初期状態では入力欄と共有ボタン、「もう一度遊ぶ」ボタンを非表示にして。

初期状態で入力欄と共有ボタン、「もう一度遊ぶ」ボタンを非表示にするには、次のようにJavaScriptのコードを変更します。

```
// 入力欄を非表示にする
inputElement.style.display = 'none';

// Twitterへの共有ボタンを非表示にする
shareButton.style.display = 'none';

// 「もう一度遊ぶ」ボタンを非表示にする
playAgainButton.style.display = 'none';
```

このようにJavaScriptのコードを変更することで、初期状態で入力欄と共有ボタン、「もう一度遊ぶ」ボタンを非表示にすることができます。

K スタートボタンが押されたら入力欄を表示するように変更して。

スタートボタンが押されたら入力欄を表示するように変更するには、次のようにJavaScriptのコードを変更します。

```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // 入力欄を表示する
  inputElement.style.display = 'block';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = theme;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
    }
  }, 1000);
});
```



30秒経過したら、入力欄とタイマーを非表示にするようにして。



30秒経過したら、入力欄とタイマーを非表示にするように変更するには、次のように  
JavaScriptのコードを変更します。



Copy code

```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // 入力欄を表示する
  inputElement.style.display = 'block';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = theme;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウント
```



コードの途中から続きを教えて。



JavaScriptでWebアプリを実装する場合、次のようなJavaScriptのコードが考えられます。



Copy code

```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // 入力欄を表示する
  inputElement.style.display = 'block';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = theme;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => [
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
      clearInterval(intervalId);

      // 入力欄を非表示にする
      inputElement.style.display = 'none';

      // タイマーを非表示にする
      timerElement.textContent = '';

      // 入力された川柳を表示する
      haikuElement.textContent = inputElement.value;

      // Twitterへの共有ボタンを表示する
      shareButton.style.display = 'inline-block';

      // 「もう一度遊ぶ」ボタンを表示する
      playAgainButton.style.display = 'inline-block';
    }
  ], 1000);
});
```



入力された川柳を共有ボタンより上にして。



入力された川柳を共有ボタンより上にするには、次のようにHTMLのマークアップを変更します。

Copy code

```
<!-- 川柳 -->
<p id="haiku"></p>

<!-- 共有ボタン -->
<button id="share">Twitterで共有する</button>
```

このようにHTMLのマークアップを変更することで、入力された川柳が共有ボタンより上に表示されるようになります。



共有ボタンと「もう1度遊ぶ」ボタンの場所を入れ替えて。



共有ボタンと「もう1度遊ぶ」ボタンの場所を入れ替えるには、次のようにHTMLのマークアップを変更します。

Copy code

```
<!-- 川柳 -->
<p id="haiku"></p>

<!-- Twitterへの共有ボタン -->
<button id="share">Twitterで共有する</button>

<!-- 「もう一度遊ぶ」ボタン -->
<button id="play-again">もう一度遊ぶ</button>
```

このようにHTMLのマークアップを変更することで、共有ボタンと「もう1度遊ぶ」ボタンの場所が入れ替わります。



これでは変わっていない。共有ボタンを「もう1度遊ぶ」ボタンより下に配置して。



共有ボタンを「もう1度遊ぶ」ボタンより下に配置するには、次のようにHTMLのマークアップを変更します。

Copy code

```
<!-- 川柳 -->
<p id="haiku"></p>

<!-- 「もう一度遊ぶ」ボタン -->
<button id="play-again">もう一度遊ぶ</button>

<!-- Twitterへの共有ボタン -->
<button id="share">Twitterで共有する</button>
```

このようにHTMLのマークアップを変更することで、共有ボタンが「もう1度遊ぶ」ボタンより下に配置されるようになります。



お題がきちんとお題として伝わるようにJavaScriptで表示を調整して。



```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // 入力欄を表示する
  inputElement.style.display = 'block';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = `お題: ${theme}`;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = remainingSeconds;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = remainingSeconds;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
      clearInterval(intervalId);

      // 入力欄を非表示にする
      inputElement.style.display = 'none';

      // タイマーを非表示にする
      timerElement.textContent = '';

      // 入力された川柳を表示する
      haikuElement.textContent = inputElement.value;

      // Twitterへの共有ボタンを表示する
      shareButton.style.display = 'inline';
    }
  }, 1000);
});
```



カウントダウンを「残り～秒」という表示になるように変更して。



カウントダウンを「残り～秒」という表示にするには、次のようにJavaScriptのコードを変更します。



```
// スタートボタンがクリックされたときの処理
startButton.addEventListener('click', () => {
  // スタートボタンを非表示にする
  startButton.style.display = 'none';

  // 入力欄を表示する
  inputElement.style.display = 'block';

  // お題をランダムに選択する
  const theme = themes[Math.floor(Math.random() * themes.length)];

  // お題を表示する
  themeElement.textContent = `お題: ${theme}`;

  // 30秒のカウントダウンを開始する
  let remainingSeconds = 30;
  timerElement.textContent = `残り${remainingSeconds}秒`;

  // 1秒ごとにカウントダウンを行う
  const intervalId = setInterval(() => {
    // 残り秒数を更新する
    remainingSeconds -= 1;
    timerElement.textContent = `残り${remainingSeconds}秒`;

    // もし残り秒数が0なら
    if (remainingSeconds === 0) {
      // カウントダウンを停止する
      clearInterval(intervalId);

      // 入力欄を非表示にする
      inputElement.style.display = 'none';

      // タイマーを非表示にする
      timerElement.textContent = '';

      // 入力された川柳を表示する
      haikuElement.textContent = inputElement.value;

      // Twitterへの共有ボタンを表示する
      shareButton.style.display = 'inline-block';
    }
  }, 1000);
});
```

```
// 「もう一度遊ぶ」ボタンを表示する
playAgainButton.style.display = 'inline-block';
},
1000);
});
```

このようにJavaScriptのコ



Twitterへ共有するとき、ツイートの内容が以下のテンプレートのようになるように変更して。



お題: {お題}

{入力された川柳}

#30秒川柳



Twitterへ共有するとき、ツイートの内容が以下のテンプレートのようになるようにするには、次のようにJavaScriptのコードを変更します。

Copy code

```
// Twitterへの共有ボタンがクリックされたときの処理
shareButton.addEventListener('click', () => {
  // Twitterへ共有する内容を作成する
  const theme = themeElement.textContent.replace('お題: ', '');
  const haiku = inputElement.value;
  const text = `お題: ${theme}\n\n${haiku}\n\n#30秒川柳`;

  // Twitterへの共有URLを作成する
  const url = 'https://twitter.com/intent/tweet?text=${encodeURIComponent(text)}';

  // Twitterへの共有URLに遷移する
  window.open(url);
});
```

このようにJavaScriptのコードを変更することで、Twitterへ共有するとき、ツイートの内容が指定したテンプレートのようになります。



CSSを追加して和風でおしゃれなデザインにして。



和風でおしゃれなデザインにするには、次のようなCSSを追加します。

Copy code

```
* {
  font-family: 'Noto Sans JP', sans-serif;
}

body {
  background-color: #e6e6e6;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}

.container {
  width: 500px;
  text-align: center;
}

.title {
  font-size: 24px;
  font-weight: bold;
  margin-bottom: 24px;
  color: #333;
}

.theme {
  font-size: 20px;
  font-weight: bold;
  margin-bottom: 12px;
  color: #333;
}

.input {
  width: 100%;
  height: 48px;
  padding: 8px;
  font-size: 16px;
  box-sizing: border-box;
  border: 2px solid #333;
  border-radius: 4px;
  outline: none;
}
```



```
.timer {
  font-size: 20px;
  font-weight: bold;
  margin-bottom: 12px;
  color: #333;
}

.haiku {
  font-size: 16px;
  margin-bottom: 24px;
  color: #333;
}

.button {
  display: inline-block;
  padding: 8px 16px;
  font-size: 16px;
  font-weight: bold;
  color: #fff;
  background-color: #333;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

.button:active {
  background-color: #555;
}
```



このようなCSSを追加することで、和風でおしゃれなデザインになります。

K HTMLの変更箇所は？



HTMLの変更箇所としては、次のような箇所があります。



- ・共有ボタンを「もう1度遊ぶ」ボタンより下に配置するために、共有ボタンと「もう1度遊ぶ」ボタンの並び順を変更する
- ・CSSを適用するために、HTMLのマークアップにクラス名を追加する

具体的には次のような変更を行います。

```
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <input class="input" id="input" type="text" placeholder="30秒で川柳を入力してください">
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku" id="haiku"></p>
  <button class="button" id="share">Twitterで共有する</button>
  <button class="button" id="play-again">もう一度遊ぶ</button>
</div>
```



このようにHTMLのマークアップを変更します。

K CSSをHTMLで読み込む方法は？



HTMLでCSSを読み込むには、次のように`<link>`要素を使用します。



```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```



このように`<link>`要素を使用することで、HTMLからCSSを読み込むことができます。

また、`<style>`要素を使用することでもHTML内にCSSを記述して読み込むことができます。

```
<head>
  <style>
    /* CSSをここに記述する */
  </style>
</head>
```



このように`<style>`要素を使用することで、HTML内にCSSを記述して読み込むことができます。



不要なスクロールが少し発生している場合、次のような原因が考えられます。



- ・ページ上部に余白がある
- ・ページ左右に余白がある
- ・ページ上部や左右に枠線が表示されている

これらの原因を確認し、必要な修正を行うことで不要なスクロールを防ぐことができます。

具体的には、次のような修正を行います。

- ・ページ上部に余白がある場合は、HTMLやbody要素にmargin: 0;というスタイルを指定する
- ・ページ左右に余白がある場合は、HTMLやbody要素にwidth: 100%;というスタイルを指定する
- ・ページ上部や左右に枠線が表示されている場合は、HTMLやbody要素にborder: 0;というスタイルを指定する

例えば、次のような修正を行うことで、不要なスクロールを防ぐことができます。



```
html,  
body {  
    margin: 0;  
    width: 100%;
```



続きを教えて。



続きは次のようにになります。



```
border: 0;  
}
```



このように修正を行うことで、不要なスクロールを防ぐことができます。

また、HTMLやbody要素にheight: 100%;というスタイルを指定することで、ページの高さが100%になるようにすることもできます。これにより、ページの高さがウインドウの高さに合わせて自動的に調整されるようになります。



```
html,  
body {  
    margin: 0;  
    width: 100%;  
    height: 100%;  
    border: 0;
```



このようにHTMLやbody要素に必要なスタイルを指定することで、不要なスクロールを防ぐことができます。



入力欄で複数行入力できるようにして、完全なHTMLを示して。



入力欄で複数行入力するには、「<input>」要素のtype属性をtextareaに変更する必要があります。



```
<textarea class="input" id="input" placeholder="30秒で川柳を入力してください">  
</textarea>
```



また、入力欄の高さを固定するために、CSSでheightプロパティを指定することもできます。



```
.input {  
    height: 48px; /* ここで高さを指定する */  
    /* 他のスタイルは同じ */  
}
```



このようにすることで、入力欄で複数行入力ができるようになります。

完全なHTMLは次のようにになります。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>30秒川柳</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1 class="title">30秒川柳</h1>
    <p class="theme" id="theme"></p>
    <textarea class="input" id="input" placeholder="30秒で川柳を入力してください"></textarea>
    <p class="timer" id="timer"></p>
    <button class="button" id="start">スタート</button>
    <p class="haiku" id="haiku"></p>
    <button class="button" id="share">Twitterで共有する</button>
    <button class="button" id="play-again">もう一度遊ぶ</button>
  </div>
  <script src="app.js"></script>
</body>
</html>
```

このようにすることで、入力欄で複数行入力ができるようになります。

K 共有ボタンと「もう1度遊ぶ」ボタンの位置を逆にして



共有ボタンと「もう1度遊ぶ」ボタンの位置を逆にするには、HTMLのマークアップを次のように変更します。

```
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <textarea class="input" id="input" placeholder="30秒で川柳を入力してください"></textarea>
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku" id="haiku"></p>
  <button class="button" id="play-again">もう一度遊ぶ</button>
  <button class="button" id="share">Twitterで共有する</button>
</div>
```

このように変更することで、共有ボタンと「もう1度遊ぶ」ボタンの位置が逆になります。

また、CSSでボタンの並び順を指定することもできます。例えば、次のようにすると、「もう1度遊ぶ」ボタンが共有ボタンより下に配置されます。

```
.button {
  /* ボタンの共通スタイルは同じ */
  order: 1;
}

#share {
  order: 2;
}
```

このようにCSSでボタンの並び順を指定することもできます。

K 複数行表示できるように入力欄の高さを変更して



入力欄の高さを変更するには、CSSでheightプロパティを指定します。例えば、次のようにすると、入力欄の高さが100pxになります。

```
.input {
  height: 100px;
  /* 他のスタイルは同じ */
}
```

このようにすることで、入力欄の高さを変更することができます。

また、高さを変更した場合は、overflowプロパティを指定することで、入力欄から文字がはみ出たときの表示方法を指定することもできます。例えば、次のようにすると、文字がはみ出たときにスクロールバーが表示されます。

```
.input {  
  height: 100px;  
  overflow: auto;  
  /* 他のスタイルは同じ */  
}
```

このようにすることで、入力欄の高さを変更し、文字がはみ出たときの表示方法を指定することができます。

**K** 入力欄の要素はInputElement、川柳を表示する要素はhaikuElementという変数に入っています。入力された川柳を表示するときに、改行文字でHTMLでも改行されて表示されるようにJavaScriptを変更して。

  入力された川柳を改行文字でHTMLでも改行されるように表示するには、JavaScriptで次のようにします。

```
// 入力された川柳を取得します  
const haiku = inputElement.value;  
  
// 入力された川柳を改行文字でHTMLでも改行されるように表示する  
// '\n'を<br>に変換するために、replace()メソッドを使います  
haikuElement.innerHTML = haiku.replace(/\n/g, '<br>');
```

このようにすることで、入力された川柳を改行文字でHTMLでも改行されるように表示することができます。

また、`<textarea>`要素では、改行文字を入力することができます。したがって、改行文字で入力された川柳は、そのまま`<textarea>`要素から取得されます。このため、上記のコードでは`\n`を`<br>`に変換しています。

また、改行文字を正規表現で表現するには、`\n`を使用します。したがって、上記のコードでは`\n/g`と書いています。`\n/g`は、文字列内のすべての改行文字を表します。また、`replace()`メソッドの第2

**K** 入力欄をリサイズできないようにして。



  入力欄をリサイズできないようにするには、CSSでresizeプロパティをnoneに指定します。例えば、次のようにすると、入力欄をリサイズできなくなります。

```
.input {  
  resize: none;  
  /* 他のスタイルは同じ */  
}
```

このようにすることで、入力欄をリサイズできないようにすることができます。

また、HTML5では、`<textarea>`要素にresize属性を指定することでも、入力欄をリサイズできないようにすることができます。例えば、次のようにすると、入力欄をリサイズできなくなります。

```
<textarea class="input" id="input" placeholder="30秒で川柳を入力してください"  
        resize="none"></textarea>
```

このようにすることで、入力欄をリサイズできないようにすることができます。

また、JavaScriptでもresizeプロパティを指定することで、入力欄をリサイズできないようにすることができます。例えば、次のようにすると、入力欄をリサイズできなくなります。

```
const inputElement = document.getElementById('input');  
  
// 入力欄をリサイズできないようにする  
inputElement.style.resize = 'none';
```

このようにすることで、入力欄をリサイズできないようにすることができます。

**K** お題をもっと考えて





お題は、次のようになるといいでしょう。



- ・空を見上げて
- ・秋の空
- ・枯れた草原
- ・朝日に照らされた山
- ・夜空に浮かぶ月
- ・春の風
- ・涙を拭う手
- ・雨が降る街
- ・歩く人々
- ・子供たちの笑顔
- ・星空を眺める二人

このようなお題を設定することで、ユーザーが川柳を書くときにインスピレーションを得ることができます。また、このようなお題を設定することで、川柳にテーマが出てくることが期待されます。

また、お題は決まっていますが、ユーザーは自由に川柳を書くことができます。したがって、ユーザーが書いた川柳はお題とは異なる内容になることがあります。そのため、お題はあくまで参考程度にとどめることが望ましいでしょう。



これらのお題を追加して。



お題をJavaScriptで実装するには、次のようにします。



```
// お題の配列
const themes = [
  '空を見上げて',
  '秋の空',
  '枯れた草原',
  '朝日に照らされた山',
  '夜空に浮かぶ月',
  '春の風',
  '涙を拭う手',
  '雨が降る街',
  '歩く人々',
  '子供たちの笑顔',
  '星空を眺める二人',
];

// お題をランダムに選ぶために、Math.random()を使います
const theme = themes[Math.floor(Math.random() * themes.length)];

// 選んだお題を表示する要素を取得します
const themeElement = document.getElementById('theme');

// 選んだお題を表示します
themeElement.innerHTML = theme;
```

このようにしておけば、JavaScriptでお題をランダムに選択して表示することができます。

また、HTMLでは、JavaScriptで選んだお題を表示する要素を用意する必要があります。例えば、次のようにすると、JavaScriptで選んだお題を表示することができます。



```
<!-- お題を表示する要素 -->
<p id="theme"></p>
```

このようにすることで、JavaScriptで選んだお題を表示することができます。

また、JavaScriptでは、Math.random()とMath.floor()を使って、配列からランダムに要素を選択しています。Math.random()は0以上1未満のランダムな数値を生成します。そのため、次のようにすると、配列の要素数よりも小さいランダムな数値を生成することができます。



お題の配列にもっとお題を追加して。





お題の配列をもっと充実させるには、次のようにします。



```
// お題の配列
const themes = [
  '空を見上げて',
  '秋の空',
  '枯れた草原',
  '朝日に照られた山',
  '夜空に浮かぶ月',
  '春の風',
  '涙を拭う手',
  '雨が降る街',
  '歩く人々',
  '子供たちの笑顔',
  '星空を眺める二人',
  '雪の空',
  '秋晴れの山々',
  '水たまり',
  '空を飛ぶ鳥',
  '風に揺れる木々',
  '満月の夜',
  '深い森',
  '花畠',
  '川を流れる水',
  '砂浜',
  '海岸',
  '山道',
  '山の麓',
  '山頂',
  '自然の美しさ',
  '僕たちの心',
];
// お題をランダムに選ぶために、Math.random()を使います
const theme = themes[Math.floor(Math.random() * themes.length)];
// 選んだお題を表示する要素を取得します
const themeElement = document.getElementById('theme');
// 選んだお
```



現在のURLを取得して、Twitterへ共有をするときに、ツイートにURLを追加するように、共有ボタンが押されたときの処理を変更して。

現在のコード:

```
...
// Twitterへの共有ボタンがクリックされたときの処理
shareButton.addEventListener("click", () => {
  // Twitterへ共有する内容を作成する
  const theme = themeElement.textContent.replace("お題: ", "");
  const haiku = inputElement.value;
  const text = `お題: ${theme}\n\n${haiku}\n\n#30秒川柳`;

  // Twitterへの共有URLを作成する
  const url = `https://twitter.com/intent/tweet?text=${encodeURIComponent(
    text
  )}`;

  // Twitterへの共有URLに遷移する
  window.open(url);
});
```



現在のURLを取得して、Twitterへ共有をするときに、ツイートにURLを追加するには、次のようになります。



```
// Twitterへの共有ボタンがクリックされたときの処理
shareButton.addEventListener("click", () => {
  // 現在のURLを取得する
  const url = window.location.href;

  // Twitterへ共有する内容を作成する
  const theme = themeElement.textContent.replace("お題: ", "");
  const haiku = inputElement.value;
  const text = `お題: ${theme}\n\n${haiku}\n\n#30秒川柳\n\n${url}`;

  // Twitterへの共有URLを作成する
  const shareUrl = `https://twitter.com/intent/tweet?text=${encodeURIComponent(
    text
  )}`;

  // Twitterへの共有URLに遷移する
  window.open(shareUrl);
});
```

このようにすることで、現在のURLを取得して、Twitterへ共有する際に、ツイートにURLを追加することができます。

また、JavaScriptでは、`window.location.href`を使って現在のURLを取得しています。さら

K 初期状態で非表示にする要素は、JavaScriptではなくCSSで非表示にして、JavaScriptとCSSの変更箇所を示して。

```
```
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <textarea
    class="input"
    id="input"
    placeholder="30秒で川柳を入力してください"
  ></textarea>
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku" id="haiku"></p>
  <button class="button" id="play-again">もう一度遊ぶ</button>
  <button class="button" id="share">Twitterで共有する</button>
</div>
```

```
// お題を表示する要素を取得する
const themeElement = document.getElementById("theme");

// 入力欄を取得する
const inputElement = document.getElementById("input");

// タイマーを表示する要素を取得する
const timerElement = document.getElementById("timer");

// スタートボタンを取得する
const startButton = document.getElementById("start");

// Twitterへの共有ボタンを取得する
const shareButton = document.getElementById("share");

// 川柳を表示する要素を取得する
const haikuElement = document.getElementById("haiku");

// 「もう一度遊ぶ」ボタンを取得する
const playAgainButton = document.getElementById("play-again");

// 入力欄を非表示にする
inputElement.style.display = "none";

// Twitterへの共有ボタンを非表示にする
shareButton.style.display = "none";

// 「もう一度遊ぶ」ボタンを非表示にする
playAgainButton.style.display = "none";
```

```

S 初期状態で非表示にする要素は、JavaScriptではなくCSSで非表示にすることができます。 ▲ ↗

JavaScriptでは、非表示にする要素を取得して、`style.display`プロパティに`"none"`を設定することで非表示にすることができます。

しかし、この方法だと、JavaScriptとHTMLとCSSが混在してしまい、HTMLとCSSが分離していないように見えます。

そこで、JavaScriptとHTMLとCSSを分離したい場合は、CSSで非表示にすることができます。

例えば、上記のHTMLのコードに対して、次のようにCSSを追加することで、初期状態で非表示にすることができます。

```
<style>
  /* 初期状態で非表示にする要素のCSS */
  .hidden {
    display: none;
  }
</style>
```

そして、HTMLを次のように変更することで、初期状態で非表示にすることができます。

```
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <textarea
    class="input hidden"
    id="input"
    placeholder="30秒で川柳を入力してください"
  ></textarea>
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku hidden" id="haiku"></p>
  <button class="button hidden" id="play-again">もう一度遊ぶ</button>
  <button class="button hidden" id="share">Twitterで共有する</button>
</div>
```

このようにすることで、JavaScriptとHTMLとCSSが分離したままで、初期状態で非表示にすることができます。

また、JavaScriptで非表示にする場合と違い、HTMLとCSSとJavaScriptが分離しているため、HTML

JavaScriptはどのように変更すれば良い？

JavaScriptは、HTMLとCSSとJavaScriptが分離したままで、初期状態で非表示にすることができます。

例えば、上記のHTMLのコードとCSSのコードに対して、次のようにJavaScriptを変更することで、初期状態で非表示にすることができます。

```
// お題を表示する要素を取得する
const themeElement = document.getElementById("theme");

// 入力欄を取得する
const inputElement = document.getElementById("input");

// タイマーを表示する要素を取得する
const timerElement = document.getElementById("timer");

// スタートボタンを取得する
const startButton = document.getElementById("start");

// Twitterへの共有ボタンを取得する
const shareButton = document.getElementById("share");

// 川柳を表示する要素を取得する
const haikuElement = document.getElementById("haiku");

// 「もう一度遊ぶ」ボタンを取得する
const playAgainButton = document.getElementById("play-again");
```

このようにすることで、JavaScriptとHTMLとCSSが分離したままで、初期状態で非表示にすることができます。

このWebアプリを開発したことを宣伝するためのバズりそうなツイートの内容を考えて。 ChatGPTがデザインから実装まですべてやった新しいWebアプリ「30秒川柳」をリリースしました！🎉 30秒で川柳を書いて、Twitterで共有できます。ぜひ、遊んでみてください。 #30秒川柳 #Webアプリ #AI #デザイン #プログラミング」

川柳を表示する要素を非表示にしないようにHTMLを修正して。JavaScriptは変更しないで。

```
...
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <textarea
    class="input hidden"
    id="input"
    placeholder="30秒で川柳を入力してください"
  ></textarea>
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku hidden" id="haiku"></p>
  <button class="button hidden" id="play-again">もう一度遊ぶ</button>
  <button class="button hidden" id="share">Twitterで共有する</button>
</div>
...
```



HTMLを修正するには、以下のようにします。

Copy code

```
<div class="container">
  <h1 class="title">30秒川柳</h1>
  <p class="theme" id="theme"></p>
  <textarea
    class="input hidden"
    id="input"
    placeholder="30秒で川柳を入力してください"
  ></textarea>
  <p class="timer" id="timer"></p>
  <button class="button" id="start">スタート</button>
  <p class="haiku" id="haiku"></p>
  <button class="button hidden" id="play-again">もう一度遊ぶ</button>
  <button class="button hidden" id="share">Twitterで共有する</button>
</div>
```

上記の修正では、入力欄、「もう一度遊ぶ」ボタン、共有ボタンの3つの要素にクラス'hidden'を追加しました。このクラスは、CSSで定義されているものとします。例えば、以下のように定義されているとします。

Copy code

```
.hidden {
  display: none;
}
```

このようにすることで、上記の3つの要素は初期状態では非表示になります。ただし、JavaScriptで非表示にする必要はありません。



スマホの表示の倍率がおかしい。修正して。



この問題はCSSで解決できます。HTMLのhead要素内に以下のmetaタグを追加すると解決で Copy code

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

このmetaタグは、HTMLのhead要素内に記述することで、Webサイトの表示倍率やレイアウトがスマートフォンなどの画面で正しく表示されるようになります。このmetaタグを追加することで、スマートフォンなどの画面でWebサイトが正しく表示されるようになります。



スマホで表示すると、bodyの高さが表示領域より大きくなつて、縦スクロールが発生する。bodyの高さを表示領域いっぱいに表示されるように修正して。



修正前のCSS

```
...
html,
body {
  margin: 0;
  width: 100%;
  border: 0;
}

body {
  background-color: #e6e6e6;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}
...
```



修正後のCSS

Copy code

```
html,
body {
  margin: 0;
  width: 100%;
  border: 0;
}

body {
  background-color: #e6e6e6;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100%;
}
```

修正する箇所は、body要素のheightプロパティです。

heightプロパティの値を100vhから100%に変更します。

100vhは、ブラウザーの表示領域の高さを表します。

一方で100%は、親要素の幅・高さを100%表すことを意味します。  
このようにすることで、body要素の高さが表示領域いっぱいに表示されるようになります。

ちなみに、このような場合は、widthプロパティも同様に100%に変更しておくと、スマートフォンでも横スクロールが発生しないようになります。

K 要素が縦の中央に表示されない。修正して。



修正後のCSS

Copy code

```
html,  
body {  
  margin: 0;  
  width: 100%;  
  border: 0;  
  height: 100%;  
}  
  
body {  
  background-color: #e6e6e6;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

修正箇所は、「body」要素に「height: 100%」を追加することにより、「body」要素の高さを表示領域いっぱいに表示させることができます。また、「html」要素にも「height: 100%」を追加しておくことで、「html」要素の高さも表示領域いっぱいになるようになります。

K 画面の左右に余白を追加して。



```
...  
html,  
body {  
  margin: 0;  
  width: 100%;  
  border: 0;  
  height: 100%;  
}  
  
body {  
  background-color: #e6e6e6;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}  
...
```

修正しました。

Copy code

```
html,  
body {  
  margin: 0;  
  width: 100%;  
  border: 0;  
  height: 100%;  
  padding: 0 16px;  
}  
  
body {  
  background-color: #e6e6e6;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

このように、HTMLとbody要素に対して、左右の余白を16pxずつ追加するようにCSSを修正することで、画面の左右に余白が追加されます。このような修正を行うことで、要素が縦横の中央に表示されるようになります。

K 横スクロールが発生した。余白を残したまま、スクロールしないように幅を修正して。



修正前のCSS

```
...
html,
body {
margin: 0;
width: 100%;
border: 0;
height: 100%;
padding: 0 16px;
}

body {
background-color: #e6e6e6;
display: flex;
align-items: center;
justify-content: center;
}
...
```

K 修正後のCSS



Copy code

```
html,
body {
margin: 0;
width: 100%;
border: 0;
height: 100%;
padding: 0 16px;
box-sizing: border-box;
}

body {
background-color: #e6e6e6;
display: flex;
align-items: center;
justify-content: center;
}
```

'box-sizing: border-box;'をbodyに追加することで、余白を含めた幅で要素が表示されるようになります。

K 画面の左右の余白が2重になっている。修正して。



修正前のCSS

```
...
html,
body {
margin: 0;
width: 100%;
border: 0;
height: 100%;
padding: 0 16px;
box-sizing: border-box;
}

body {
background-color: #e6e6e6;
display: flex;
align-items: center;
justify-content: center;
}
...
```

K 修正後のCSS



Copy code

```
html,
body {
margin: 0;
width: 100%;
border: 0;
height: 100%;
box-sizing: border-box;
}

body {
padding: 0 16px;
background-color: #e6e6e6;
display: flex;
align-items: center;
justify-content: center;
}
```

以上の修正により、横スクロールがなくなり、画面の左右に余白ができ、要素が縦の中央に表示されるようになります。

