

# じゃんけんで勝負！

■作成した Website のスクリーンショット



## じゃんけんで勝負！

手を選んでね！



グー



チョキ



パー

あなた: グー / コンピュータ: パー

あなたの負け...

勝ち: 0 / 負け: 1 / あいこ: 0

## じゃんけんで勝負！

手を選んでね！



グー



チョキ



パー

あなた: チョキ / コンピュータ: チョキ

あいこ！

勝ち: 0 / 負け: 2 / あいこ: 1



## ■ソースコード

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8" />
  <title>じゃんけんゲーム</title>
  <style>
    body {
      font-family: "Hiragino Kaku Gothic ProN", sans-serif;
      background: #66ccff;
      text-align: center;
      padding: 2em;
    }

    h1 {
      color: #ffffff;
    }

    h2 {
      color: #ffffff;
      font-size: 1.4em;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1>じゃんけん勝負！</h1>
  <h2>手を選んでね！</h2>
  <div>
    <img alt="グー (Rock)" data-bbox="295 208 412 290"/>
    <img alt="チョキ (Scissors)" data-bbox="438 208 555 290"/>
    <img alt="パー (Paper)" data-bbox="581 208 698 290"/>
  </div>
  <div>
    あなた: パー / コンピュータ: グー
    あなたの勝ち！！
    勝ち: 1 / 負け: 2 / あいこ: 1
  </div>
</body>
</html>
```

```
}
```

```
.hand-container {  
  display: flex;  
  justify-content: center;  
  gap: 2em;  
  margin-bottom: 1em;  
}
```

```
.hand-button {  
  width: 150px;  
  height: 150px;  
  border: none;  
  background: #fff;  
  border-radius: 16px;  
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);  
  transition: transform 0.2s;  
  cursor: pointer;  
  padding: 0;  
}
```

```
.hand-button:hover {  
  transform: scale(1.05);  
}
```

```
.hand-button img {  
  width: 100%;  
  height: 100%;  
  object-fit: contain;  
  border-radius: 16px 16px 0 0;  
}
```

```
.hand-label {  
  display: block;  
  margin-top: 0.3em;  
  font-size: 1.1em;
```

```
    color: #333;
  }

  #result {
    font-size: 1.3em;
    margin-top: 1em;
    color: #ffffff;
  }

  #score {
    margin-top: 0.5em;
    font-weight: bold;
    color: #ffffff;
  }
</style>
</head>
<body>
  <h1>じゃんけんて勝負！</h1>
  <h2 style="color: #ffffff; font-weight: normal;">手を選んでね！</h2>
  <div class="hand-container">
    <div>
      <button class="hand-button" onclick="play('グー')">
        
      </button>
      <span class="hand-label">グー</span>
    </div>
    <div>
      <button class="hand-button" onclick="play('チョキ')">
        
      </button>
      <span class="hand-label">チョキ</span>
    </div>
    <div>
      <button class="hand-button" onclick="play('パー')">
        
      </button>
    </div>
  </div>

```

```

        <span class="hand-label">>パー</span>
    </div>
</div>

<div id="result"></div>
<p id="score">勝ち: 0 / 負け: 0 / あいこ: 0</p>

<script>
    const hands = ['グー', 'チョキ', 'パー'];
    let win = 0, lose = 0, draw = 0;

    function play(player) {
        const computer = hands[Math.floor(Math.random() * 3)];
        let message = `あなた: ${player} / コンピュータ: ${computer}<br>`;

        if (player === computer) {
            message += 'あいこ!';
            draw++;
        } else if (
            (player === 'グー' && computer === 'チョキ') ||
            (player === 'チョキ' && computer === 'パー') ||
            (player === 'パー' && computer === 'グー')
        ) {
            message += 'あなたの勝ち!!';
            win++;
        } else {
            message += 'あなたの負け... ';
            lose++;
        }
    }

    document.getElementById('result').innerHTML = message;
    document.getElementById('score').textContent =
        `勝ち: ${win} / 負け: ${lose} / あいこ: ${draw}`;
}
</script>
</body>

```

</html>

## ■どのような Website を作ろうとしたのか、またそのための工夫について

今回は「じゃんけんゲーム」の Web サイトを制作した。クリックで簡単に遊べるミニゲームを通して、JavaScript によるイベント処理や条件分岐、乱数処理の理解を深めることを目的とした。デザイン面では背景を明るい水色にしてポップな印象を持たせ、文字色はすべて白で統一することで読みやすくした。また、手の選択は画像付きの大きめのボタンで行えるようにし、視覚的にも楽しめるように工夫した。スコアもリアルタイムに表示されるようにし、ユーザーが自分の成績を確認しながら繰り返し遊べる構成にした。

## ■ソースコードにおける各処理ごとの解説

### <HTML>

- <h1>でページのタイトル「じゃんけんで勝負！」を表示。
- <h2>で「手を選んでね！」というプレイヤーへの案内を表示。
- .hand-container の中に、プレイヤーが選べる「グー」「チョキ」「パー」のボタンを配置。各ボタンは<button>タグ内に<img>（画像）を埋め込んで視覚的にわかりやすく。
- 各手の下には<span class="hand-label">を使い、手の名前を文字でも表示。
- <div id="result">は、プレイ後に結果（勝ち・負け・あいこ）を表示する領域。
- <p id="score">は、勝敗の合計（勝ち・負け・あいこの回数）を表示するための要素。

### <CSS>

- body で背景色を#66ccff（明るい水色）に設定し、全体にポップで楽しい印象を持たせた。
- h1, h2, #result, #score などの文字はすべて白（#ffffff）に統一し、背景とのコントラストを高めて読みやすくした。
- .hand-button は大きめの正方形で、白い背景に影と角丸をつけ、立体的に見せている。ホバー時に少し拡大することで、押しやすくなるようアニメーションも追加している。
- .hand-label のテキストには margin-top を設定し、ボタンと文字の間隔を整えている。

### <JavaScript>

JavaScript では、じゃんけんのゲームロジックを実装しています。

- const hands = ['グー', 'チョキ', 'パー'];  
→ プレイヤーとコンピュータが選ぶ「手」を定義した配列。

- `let win = 0, lose = 0, draw = 0;`  
→ 勝ち・負け・あいこの回数を記録するための変数。
- `function play(player) { ... }`  
→ プレイヤーがどれかの手を選んだときに呼び出される関数。

関数内の主な処理は以下のとおり：

1. `Math.random()`を使って、コンピュータの手をランダムに決定
  2. プレイヤーの手とコンピュータの手を比較して勝敗を判定
  3. 勝敗に応じてメッセージを生成し、`#result`に表示
  4. 各結果に応じて `win, lose, draw` のカウントを増やす
  5. スコアを `#score` にテキストで更新
- `document.getElementById(...)` を使って、HTML の要素に動的に結果を反映

## ■生成プロンプト

・JavaScript を使って何らかの Website を作成せよ。ただし、プログラムの行数は 30 行以上とする。という課題がでました。案を考えるのを手伝ってください。

### ・1. 変数

#### 1.1 変数名の付け方

#### 1.2 変数の定義

#### 1.3 データ型

##### 1.3.1 数値

##### 1.3.2 文字列

##### 1.3.3 論理値

##### 1.3.4 null 値

##### 1.3.5 undefined

### 2. 演算子

#### 2.1 算術演算子

#### 2.2 代入演算子

#### 2.3 比較演算子

#### 2.4 論理演算子

### 3. 条件分岐

#### 3.1 if 文

#### 3.2 if～else 文

#### 3.3 switch 文

### 4. 繰り返し処理

#### 4.1 for 文



## 4.2 while 文

## 5. 配列

### 5.1 配列

### 5.2 連想配列

## 6. 関数

## 7. オブジェクト

### 7.1 オブジェクトの取り出し

### 7.2 オブジェクトの作成

## 8. 正規表現

### 8.1 パターンマッチング

今回の授業で学んだのは上記のものです。これを活用できるのはありますか？

- ・じゃんけんゲームをつくりたいです。
- ・画像付きにしたいです。画像は大きめで正方形の形のボタンにして、その下にグーなどの文字を入れたいです。画面が楽しい感じになるようにしたいです。
- ・手のボタンの上に「手を選んでね！」という文字を載せたいです
- ・スコアがリアルタイムに表示されるようにしたいです。
- ・背景の色を#66ccffにかえたいです。
- ・楽しい色の組み合わせってどんなものがありますか？背景と文字の色をマッチした感じにしたいです。
- ・提案されたものが見にくかったので、背景の水色は変えず、文字はすべて白にしました。

## ■生成 AI の解答に対する評価

私の意図に合った提案をしてくれていたので良かった。