ストップウォッチ＋記録

■成果物の概要

本課題では、JavaScript を用いて「ストップウォッチ＋記録機能付き Web サイト」を制作した。開始、停止、リセット、記録(ラップ)といった基本操作に加え、記録を削除できる機能も備えている。操作に応じて背景色や状態表示が変化することで、ユーザーに視覚的なフィードバックを与える使用になっている。

【図 1】作成した Web サイト

■生成プロンプト

以下は、ChatGPT を活用して実装内容を改善・拡張した際の代表的なプロンプトである。

- 第 3 回で学んだ 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。案を一緒に考えてください。 第 3 回で学んだことは、1. ページの背景の色を変える場合 2. DOM 3. CSS の操作 4. イベント 5. 動きのあるボタン（テキストの追加）を作成する場合 6. JavaScript からのイベントの監視 7. 動的な要素の追加 8. ストップウォッチを作成する場合 8.1 JavaScript を外部ファイルとして読み込む 8.2 開始ボタンの処理 8.3 停止ボタンの処理 8.4 リセットボタンの処理 8.5 経過時刻の計算 です。
- ストップウォッチにメモ機能をつけたサイトを作りたいです
- 開始と停止からイメージされる色ってありますか？毎回色が変わるわけではなく、開始のときはこの色、停止のときはこの色、って決まった感じにしたいです
- もう少し落ち着いた色にしたいです
- 「現在どの状態か」を上にテキストで表示したいです。例えば「▶ 計測中, ■ 停止中」

みたいな

- ストップウォッチの画面をもう少し大きくしたいです。加えて、目立ちやすいように白枠の中に入れたいです。
- 開始などのボタンは横一列がいいです。 また、メモの部分が枠内に収まっていないように見えるのでもう少しゆとりをもった大きさで枠をつくりたいです。あと、メモの部分は立体感はなくていいです
- メモを取る前にの表示が細長い枠で表示されるのですが、最初から幅のある枠が表示されていたいです
- 中に表示されるメモの内容を枠の中央揃えで載せたいのですがどうしたらいいですか
- 記録したときに、メモ〇：みたいな表示がでるので、その左にある数字はなくていいのですがどこを消したらいいですか？
- 記録の削除ボタンを作りたいです

これらのプロンプトに対して、生成ＡＩは段階的にコード修正案を提示し、見た目や機能の改善に貢献した。とくに、「削除ボタンの追加」に関しては、要素の生成と削除に必要なDOM操作コードを正確に提案してくれたため、実装の理解と効率化の両面で非常に有用だった。

■ソースコードの説明

このWebサイトは、HTMLとCSSで基本的な構造と見た目を整え、JavaScriptを用いて動的な機能を実現している。主な機能は、ストップウォッチの表示更新、開始・停止・リセット・記録(ラップ)処理、記録の削除、背景色や状態テキストの変化などである。以下に主要な処理ブロックについて説明する。

- **ストップウォッチの時間更新処理**

```
function updateTime() {
  const now = Date.now() - startTime + elapsed;
  const min = String(Math.floor(now / 60000)).padStart(2, "0");
  const sec = String(Math.floor((now % 60000) / 1000)).padStart(2, "0");
  const ms = String(Math.floor((now % 1000) / 10)).padStart(2, "0");
  display.textContent = `${min}:${sec}.${ms}`;
  timerId = requestAnimationFrame(updateTime);
}
```

この関数は、`requestAnimationFrame()`を用いて約60fpsでループしながら、経過時間を計算し、ストップウォッチの画面に表示する。`padStart()`を使って常に桁数をそ

ろえたフォーマットで表示している。

- **開始・停止・リセットボタンのイベント処理**

```
document.getElementById("start").addEventListener("click", () => {
  startTime = Date.now();
  updateTime();
  document.body.style.backgroundColor = '#cce3dc';
  status.textContent = "▶ 計測中";
});
```

開始ボタンでは、startTime に現在時刻を記録し、更新処理を開始する。また、背景色や状態テキストを更新して視覚的にも計測中であることを伝えている。

```
document.getElementById("stop").addEventListener("click", () => {
  cancelAnimationFrame(timerId);
  elapsed += Date.now() - startTime;
  document.body.style.backgroundColor = '#e8c1c5';
  status.textContent = "■ 停止中";
});
```

停止ボタンでは、cancelAnimationFrame()を使ってループを停止し、経過時間を elapsed に加算して一時停止の状態を保持する。背景色と状態表示も変更する。

```
document.getElementById("reset").addEventListener("click", () => {
  cancelAnimationFrame(timerId);
  elapsed = 0;
  display.textContent = "00:00.00";
  laps.innerHTML = "";
  lapCount = 0;
  document.body.style.backgroundColor = '#d6eaf8';
  status.textContent = "■ 停止中";
});
```

リセットボタンでは、時間を初期化し、画面上の表示時間と記録リストをリセットする。背景色も初期状態に戻す。

- **記録(ラップ)ボタンと削除機能の実装**

```
document.getElementById("lap").addEventListener("click", () => {
  lapCount++;
  const li = document.createElement("li");
  li.textContent = `記録${lapCount}：${display.textContent}`;
```

```
  const delBtn = document.createElement("button");
  delBtn.textContent = "削除";
  delBtn.style.marginLeft = "10px";
  delBtn.style.fontSize = "0.9em";
  delBtn.style.padding = "2px 6px";

  delBtn.addEventListener("click", () => {
    li.remove();
  });

  li.appendChild(delBtn);
  laps.appendChild(li);

  document.body.style.backgroundColor = '#fff9c4';
});
```

ラップボタンでは、現在の表示時間を元に<li>要素を作成し、記録としてリストに追加する。また、その記録の横に「削除」ボタンを自動で作成し、クリックすると該当の記録だけを削除できるようにしている。これにより、ユーザーが必要な記録だけを保持しやすくなっている。

- **DOM 操作の活用**

  全体を通して、getElementById()による要素の取得、createElement()・appendChild()による要素の生成と追加、textContent による内容の書き換え、style.backgroundColor による背景色の変更など、多数の DOM 操作を行っている。これらにより、ユーザーの操作に応じて Web ページの見た目や内容が動的に変化する仕組みを構築した。

■工夫した点

①状態を色とテキストで明示し、視覚的な分かりやすさを向上

ストップウォッチの状態(計測中・停止中など)をユーザーが直感的に把握できるよう、画面の背景色を動的に変更する仕様にした。例えば、計測中は爽やかなミントグリーン、停止中は落ち着いたローズグレイなど、色彩心理に基づいた配色を用いることで、状態が自然に伝わるよう配慮した。

また、テキストによる状態表示(▶計測中、■停止中)も併用し、視覚的な演出と情報の正

確性を両立させている。これにより、ユーザーが現在の状態を見誤ることなく操作を進められるようにした。

【図 2】開始の画面(▶計測中)

【図 3】停止の画面(■停止中)

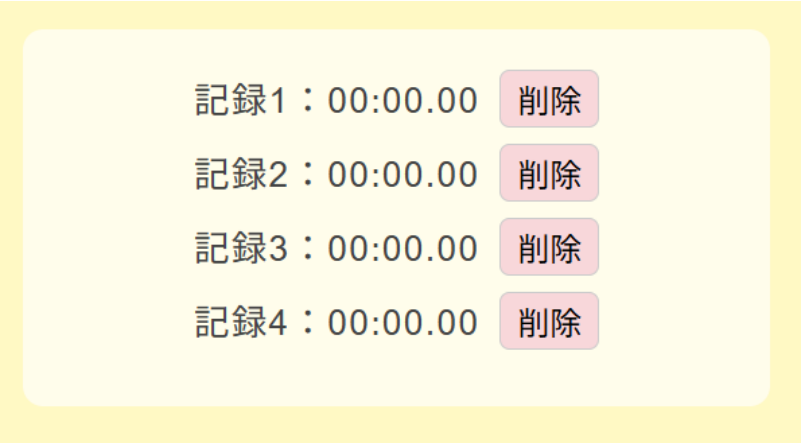
【図 4】リセットの画面

【図 5】ラップの画面

②記録(ラップ)機能の追加と、削除による柔軟な管理

ストップウォッチにおいて「ラップ」ボタン(記録)を押すことで、経過時間をメモとして記録できる機能を実装した。これにより、ユーザーが複数のタイミングを記録し比較することが可能となる。

さらに、記録ごとに削除ボタンを生成し、任意の記録だけを削除できるようにすることで、不要な記録を整理できる柔軟なインターフェースを実現した。記録と削除ボタンは DOM 操作によって動的に追加されるようにしており、必要なときだけ必要な要素が表示される構成になっている。



【図 6】削除ボタン

③レイアウト・デザイン面での配慮

ユーザーがストレスなく操作できるよう、ストップウォッチの表示領域を大きくし、中央に配置することで視認性を高めた。白背景＋グレー枠による落ち着いた表示に加え、ボタンも等間隔で横並びにし、モバイル環境でも使いやすいようレスポンシブ対応のスタイルを取り入れている。

また、記録リストはあらかじめ十分な幅と高さを確保し、どのような文字数や回数の記録でも、レイアウトが崩れにくいように設計している。記録の表示は中央揃えにし、情報がきれいに整列するように意識した。