



対象者視点でのワークショップ体験シミュレーション



想定対象者プロフィール

- 年齢：高専1-2年生
- スキル：HTML/CSS/JavaScript基礎学習済み
- 経験：簡単な静的ページは作ったことがある
- 心理状態：「動的なサイトを作りたい」けど「どこから手をつけていいかわからない」



Phase 0: 完成形体験・分析（10分）

対象者の反応シミュレーション



Step 0-1: デモ体験（5分）

対象者の心境：

- 「おお、普通にきれいなサイトだ」
- 「チェックボックスをクリックしたら進捗バーが動いた！」
- 「ホバーしたら浮き上がる感じになる、かっこいい」
- 「全部チェックしたらメッセージが出た！」

つまづきポイント：

- 特になし（体験のみなので）



Step 0-2: 機能分解（5分）

対象者の思考：

- 「ええと、上にタイトルがあって...」
- 「進捗バーってどうやって作るんだろう？」
- 「チェックボックスは知ってるけど、こんなにきれいにできるの？」

つまづきポイント：

- 進捗バーの実装方法が想像つかない
- 「どこがHTMLでどこがCSSなのかわからない」

必要なサポート：

- 「今は実装方法を考えなくて大丈夫、まずは何があるかだけ見つけましょう」
- ホワイトボードに要素を書き出してもらう

 Phase 1: HTML構造実装（30分）**対象者の体験シミュレーション** **Step 1-1: 要素の洗い出し（10分）****対象者の思考プロセス：**

「ええと、タイトル部分は...h1タグ？」
「進捗バーは...divかな？でもどうやって棒にするんだろう」
「チェックボックスはinputタグで...」
「ラベルはlabelタグ？でもこんなにきれいになるの？」

つまづきポイント：

- 進捗バーの構造が想像できない
- 「classとidどちらを使えばいい？」
- 「セマンティックHTMLって何？」

必要なサポート：

- 「進捗バーは外側のdivと内側のdivの2重構造ですよ」
- 「classは複数の要素、idは1つの要素に使います」
- 「headerタグ、sectionタグを使うと意味的に正しいHTMLになります」

 Step 1-2: HTML実装（20分）**対象者の実装体験：**

5分経過時点：

```
<h1>今日の学習タスク</h1>
<p>目標：CSS装飾とJavaScript動きをマスターしよう！</p>
<!-- 「あれ、header タグを使った方がいいって言ってたな」 -->
```

10分経過時点：

```
<header>
  <h1><img alt="rainbow icon" data-bbox="85 268 105 283"/> 今日の学習タスク</h1>
  <p>目標：CSS装飾とJavaScript動きをマスターしよう！</p>
</header>

<div>
  <h2><img alt="checkmark icon" data-bbox="85 351 105 366"/> 学習進捗</h2>
  <div>進捗バー</div>
  <!-- 「進捗バーってどう書くんだっけ？」 -->
</div>
```

つまづきポイント：

- 進捗バーの2重div構造を忘れる
- class名/id名の命名に迷う
- 「完了メッセージはどこに書けばいい？」

必要なサポート：

- 「進捗バーは `<div class="progress-bar"><div class="progress-fill"></div></div>` です」
- 「class名は後でCSSで使うので、わかりやすい名前をつけましょう」
- 「完了メッセージは最初は非表示なので、body の最後でOKです」

20分経過時点（期待する状態）：

- 基本的なHTML構造は完成
- 見た目はまだブラウザのデフォルトスタイル

 Phase 2: CSS基本装飾（35分）

対象者の体験シミュレーション

Step 2-1: 完成形との見た目比較 (5分)

対象者の反応：

- 「うわ、全然違う！」
- 「背景がグレーで、白いカードみたいになってる」
- 「フォントも違うし、間隔も全然違う」

発見するギャップ：

- 「背景色がない」
- 「カードの影がない」
- 「余白が全然足りない」

Step 2-2: 基本レイアウトCSS (25分)

5分経過時点（全体設定）： 対象者の思考：

- 「 って何？全部に適用される？」
- 「box-sizing: border-box って何だっけ？」

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
body {  
  background: #f3f4f6;  
  font-family: sans-serif;  
}
```

つまづきポイント：

- リセットCSSの意味がわからない
- 「色コードってどうやって決めるの？」

15分経過時点（カードレイアウト）： 対象者の困惑：

- 「border-radius で角丸になるのはわかるけど…」
- 「box-shadow の数値の意味がわからない」

```
.main-header, section {
  background: white;
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}
```

つまづきポイント：

- box-shadow の構文が複雑
- 「rgbaって何？」
- 「どの値がどの効果なのかわからない」

必要なサポート：

- 「box-shadow は 水平 垂直 ぼかし 色 の順番です」
- 「rgba は透明度付きの色指定です」
- 実際に値を変更して効果を確認してもらう

25分経過時点（進捗バー・タスクアイテム）： 対象者の実装：

```
.progress-bar {
  background: #e5e7eb;
  height: 25px;
  border-radius: 12px;
}

.progress-fill {
  background: #10b981;
  height: 100%;
  width: 0%; /* 「なんで0%？」 */
  border-radius: 12px;
}
```

つまづきポイント：

- 「width: 0% の意味がわからない」
- 「タスクアイテムのflexboxレイアウトが難しい」

Step 2-3: 完成形チェック（5分）

対象者の感想：

- 「おお、だいぶ似てきた！」

- 「でもまだホバーエフェクトがない」

🚀 Phase 3: CSSアニメーション (25分)

対象者の体験シミュレーション

🚀 Step 3-1: アニメーション要件分析 (5分)

対象者の発見：

- 「マウスを乗せると浮き上がる感じになる」
- 「進捗バーが幅変化してた」
- 「完了メッセージがポップアップで出る」

🧑🏻 Step 3-2: ホバーエフェクト実装 (15分)

5分経過時点： 対象者の試行錯誤：

```
.task-item:hover {  
  /* 何を書けばいいんだろう？ */  
}
```

つまづきポイント：

- 「:hover は知ってるけど、何を変化させればいい？」
- 「浮き上がる感じってどうやって作るの？」

10分経過時点：

```
.task-item:hover {  
  background: #f1f5f9;  
  /* 「色は変わったけど、浮き上がらない」 */  
}
```

必要なサポート：

- 「transform: translateY(-2px) で上に移動できます」
- 「transitionを使うとなめらかに変化します」

15分経過時点：

```
.task-item {
  transition: all 0.3s ease;
}

.task-item:hover {
  background: #f1f5f9;
  border-color: #4f46e5;
  transform: translateY(-2px);
  box-shadow: 0 4px 12px rgba(79, 70, 229, 0.2);
}
```

対象者の感動：

- 「おお！浮いた！すげー！」

Phase 4: JavaScript動的機能（20分）

| 対象者の体験シミュレーション

Step 4-1: 動的機能の特定（5分）

対象者の分析：

- 「チェックすると線が引かれる」
- 「進捗バーが伸びる」
- 「全部チェックするとメッセージが出る」

Step 4-2: JavaScript実装（15分）

3分経過時点（要素取得）：

```
const checkboxes = document.querySelectorAll('.task-checkbox');
// 「querySelectorAllってquerySelectorと何が違うんだっけ？」
```

つまづきポイント：

- querySelectorとquerySelectorAllの違い

- 「NodeListって配列？」

7分経過時点（イベント処理）：

```
checkboxes.forEach(checkbox => {  
  checkbox.addEventListener('change', function() {  
    // 「ここで何をすればいいんだっけ？」  
  });  
});
```

対象者の困惑：

- 「forEachって何？」
- 「changeイベントって何？clickじゃダメ？」
- 「function()の中に何を書けばいい？」

必要なサポート：

- 「forEachは配列の各要素に対して処理を実行します」
- 「changeはチェック状態が変わった時、clickは押した時です」
- 「まずはconsole.log('クリックされた')と書いてみましょう」

15分経過時点（進捗計算）：

```
function updateProgress() {  
  const totalTasks = checkboxes.length;  
  const completedTasks = document.querySelectorAll('.task-checkbox:checked').length;  
  // 「:checkedって何？」  
  const percentage = Math.round((completedTasks / totalTasks) * 100);  
  // 「Math.roundって何だっけ？」  
}
```

つまづきポイント：

- CSS セレクタの:checked
- 計算式の理解
- DOM操作でのstyle変更

最終的な対象者の感想：

- 「動いた！でも複雑だな...」
- 「JavaScriptって難しいけど面白い」

 全体的なつまづきポイントまとめ

高頻度でつまづく部分

1. CSS の box-shadow, rgba構文
2. JavaScript のquerySelectorAll, forEach
3. 進捗バーの2重div構造
4. transform, transition の概念

必要なサポート戦略

1. 実際に値を変更して効果確認
2. console.logでデバッグ習慣
3. ペア作業での相互サポート
4. 段階的な目標設定

学習効果を高めるポイント

1. 完成形との比較を頻繁に行う
2. 動作確認のタイミングを明確に
3. 「なぜそうするのか」の説明
4. 成功体験の積み重ね