

レポート提出票

科目名: 情報工学実験3

実験課題名: 教育システム開発

実施日: 2023年 5月 24日

学籍番号: 4619027

氏名: キムドヒョン

共同実験者: 井元 将太郎 柏尾 大介

常盤木 翔大 三輪 香織

1 実験の要旨

一回目は現代テスト理論を仮想環境でシミュレーションをして行った。2回目では、古典テスト理論を利用して Web 環境のテストアプリケーションを作成した。今回の実験では2回目の実験装置に1回目の理論を埋め込むことだけである。つまり、現代テスト理論である項目反応理論をテストシステムに実装し、受験者の能力値を測る。

2 実験の目的

項目反応理論を2回目の実験で作った古典テストアプリケーションの実環境に適用して受験者の能力値を測定する。

3 実験の理論

3.1 システム

2回目の実験で作成したシステムは正答率=能力だった。このシステムに項目反応理論を適用するために、一回目の実験で利用した `estimation()` 関数を用いた。`estimation()` 関数は引数に `x` と `itemBank` が設定されているため、実用の際にはそれらを用意しなければならない。そこで、`exam` の中に受験者の回答済項目系列を意味する `bank` という項目を追加する。また、`continueTesting()` 関数の中で `estimation()` 関数を HTML に適用するため `3.functions.js` だけでなく `1.functions.js` も読み込む必要がある。

3.2 インタフェース

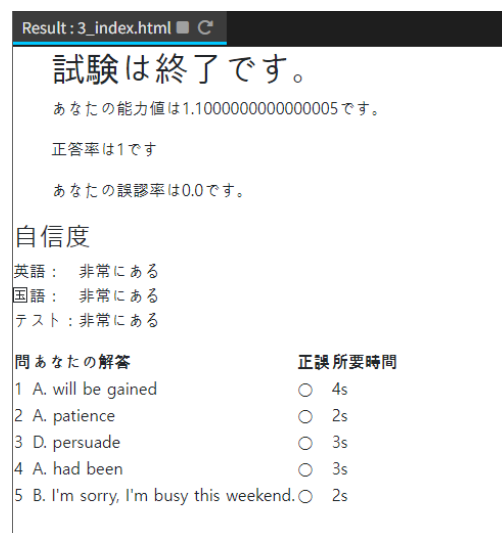
インタフェースのデザインについて今回は Bootstrap という Twitter 風のデザインツールキットを用いて CDN(Content Delivery Network) を介して HTML に適用する。CDN は全国に散らばった専用のサーバからユーザへ最短経路でライブラリを配信する技術であり、ツールをインストールするよりも手軽である。

2回目までのシステムは、無回答時にも次の問題に進むが、今回の実験では「受験者の無回答を許容せず、絶対に何かしらの項目をしらの項目を選択させる」と定める。これを実装するためには、HTML 標準の `required` を用いる。また、テキスト部分のクリックによりラジオボタンがチェックができるように `for` 属性を入れる。

4 課題

- 演習 3-1- 逐次的に能力値を推定する機能を実装し、最後に能力値が表示されるか確認しなさい。
- 演習 3-2- インタフェース改善のため、「受験者の無回答は許容せず、絶対に何かしらの項目をしらの項目を選択させる」「テキストの部分のクリックでラジオボタンの選択が可能」の2つの仕様を、これまでに開発してきたテストシステムに組み込みなさい。
- 課題 3-1- 必要な機能を実装した上で前頁の実験を実施し、各受験者の能力値（項目反応理論）と正答率（古典テスト理論）、および主観評価の結果を記録しなさい。また、能力値と主観評価とのずれを、正答率との比較により考察しなさい。
- 課題 3-2- 受験者の能力値と回答反応（正答率・各問題の正誤は不可）の相関係数を算出し、問題項目パラメタ（自分の利用した具体的なパラメタについて行うこと。aパラ、bパラの両方）と関連づけて考察しなさい。
- 課題 3-3- 本実験で開発したテストシステムの改善案について、1. 能力値推定の観点から、2. インタフェースの観点から、それぞれ考察しなさい。
改善案の仕様図もしくはデザインイメージを、少なくとも一枚作成すること。

4.1 演習 3-1



Result: 3_index.html

試験は終了です。

あなたの能力値は1.1000000000000005です。

正答率は1です

あなたの誤謬率は0.0です。

自信度

英語： 非常にある
国語： 非常にある
テスト：非常にある

問あなたの解答	正誤所要時間
1 A. will be gained	<input type="radio"/> 4s
2 A. patience	<input type="radio"/> 2s
3 D. persuade	<input type="radio"/> 3s
4 A. had been	<input type="radio"/> 3s
5 B. I'm sorry, I'm busy this weekend.	<input type="radio"/> 2s

図 1: 演習 3-1 の実装

4.2 演習 3-2

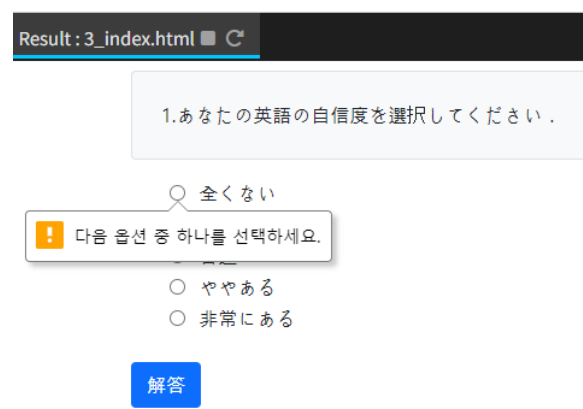


図 2: 演習 3-2 の実装

4.3 課題 3-1

<div>Re: D : 4619027</div> <div>2023年 06月 01日(木曜日) 16:20 - 4621065 常盤木 翔大 の投稿</div> <div>試験は終了です。 あなたの能力値は0.8000000000000003です。</div> <div>正答率は0.8です</div> <div>あなたの誤謬率は0.2です。</div> <div>自信度 英語： ややある 国語： ややある テスト： ややある</div> <div>問 あなたの解答 正誤 所要時間 1 A. will be gained ○ 14s 2 B. skill × 13s 3 D. persuade ○ 10s 4 A. had been ○ 21s 5 B. I'm sorry, I'm busy this weekend. ○ 12s</div>	<div>Re: D : 4619027</div> <div>2023年 06月 01日(木曜日) 16:25 - 4621015 井元 将太郎 の投稿</div> <div>あなたの能力値は-0.2999999999999998です。</div> <div>正答率は0.2です</div> <div>あなたの誤謬率は0.8です。</div> <div>自信度 英語： あまり無い 国語： 普通 テスト： あまり無い</div> <div>問 あなたの解答 正誤 所要時間 1 C. has gained × 26s 2 B. skill × 56s 3 C. tempt × 10s 4 B. have been × 9s 5 B. I'm sorry, I'm busy this weekend. ○ 12s</div>
<div>Re: D : 4619027</div> <div>2023年 06月 01日(木曜日) 16:22 - 4621088 三輪 香織 の投稿</div> <div>試験は終了です。 あなたの能力値は0.8000000000000003です。</div> <div>正答率は0.8です</div> <div>あなたの誤謬率は0.2です。</div> <div>自信度 英語： ややある 国語： 全くない テスト： 普通</div> <div>問 あなたの解答 正誤 所要時間 1 A. will be gained ○ 16s 2 分からない × 22s 3 D. persuade ○ 31s 4 A. had been ○ 38s 5 B. I'm sorry, I'm busy this weekend. ○ 33s</div>	<div>Re: D : 4619027</div> <div>2023年 06月 01日(木曜日) 16:35 - 4621033 柏尾 大介 の投稿</div> <div>試験は終了です。 あなたの能力値は0.8000000000000003です。</div> <div>正答率は0.8です</div> <div>あなたの誤謬率は0.2です。</div> <div>自信度 英語： あまり無い 国語： 普通 テスト： 普通</div> <div>問 あなたの解答 正誤 所要時間 1 A. will be gained ○ 22s 2 C. advice × 2s 3 D. persuade ○ 1s 4 A. had been ○ 2s 5 B. I'm sorry, I'm busy this weekend. ○ 2s</div>

図 3: 共同実験者の能力値と正答率

能力値（項目反応理論）は、各受験者の能力やスキルを数値化する評価方法である。項目反応理論では、受験者の応答パターンや難易度などの情報を統計的に分析し、受験者の能力値を推定する。

一方、正答率（古典テスト理論）は、受験者が正解した問題の割合を計算する評価方法です。正答率は、受験者がどれだけ問題に正解したかという結果を示すが、能力の詳細な情報や相対的な能力の比較を提供することはない。

図3の受験者の結果から見れば、主観評価を「普通」や「ややある」と選択した受験者は能力値が0.8000000000000003であって、その受験者たちは2番問題だけを間違った。正答率は1問だけ間違っただけのため、0.8であった。一方、1から4問までを間違った受験者の主観評価は「あまりない」であって、能力値は-0.2999999999999998、正答率は0.2であった。

自信度の低い人は結果的に正答率が低く、能力値も他の人より低かった。他の人は全員同じ問題を間違ってしまったため正確ではないが、自信度が高いほど正答率が高く得られるが、同じ正答率であっても能力値は問題のパラメタ（識別力や困難度）によって大きく変わると思われる。

4.4 課題3-2

1. 問題1の相関係数の算出

- 能力値データ：[0.8,0.8,-0.3,0.8]
- 回答反応データ：[14,16,26,22](所要時間)
- 相関係数の算出：-0.7868（ピアソン相関係数）
- a パラ：0.275
- b パラ：-0.507

2. 問題2の相関係数の算出

- 能力値データ：[0.8,0.8,-0.3,0.8]
- 回答反応データ：[13,22,56,2](所要時間)
- 相関係数の算出：-0.9364
- a パラ：0.364
- b パラ：1.269

3. 問題3の相関係数の算出

- 能力値データ：[0.8,0.8,-0.3,0.8]
- 回答反応データ：[10,31,10,1](所要時間)
- 相関係数の算出：0.1571
- a パラ：0.761
- b パラ：0.361

4. 問題4の相関係数の算出

- 能力値データ：[0.8,0.8,-0.3,0.8]
- 回答反応データ：[21,38,9,2](所要時間)
- 相関係数の算出：0.3595
- a パラ：0.524

- b パラ：0.086

5. 問題5の相関係数の算出

- 能力値データ：[0.8,0.8,-0.3,0.8]
- 回答反応データ：[12,33,12,2](所要時間)
- 相関係数の算出：0.1405
- a パラ：1.466
- b パラ：-0.622

各問題の回答時間と能力値の相関係数を求めた。1, 2問だけみると相関がありそうだが、3, 4, 5問の結果はそうでない。普遍的に考えられるのは、能力値が高ければ高いほど、テストの回答時間が短くなることである。実際に難易度の一番高い問題2の総回答時間が他の問題より大きかった。また、全員答えを間違った。

しかし、計算結果では1, 2問を除いて確かな負の相関を得られなかった。その理由として、ある問題に対してざっと解いて提出する人もいるからだと思われる。

4.5 課題3-3

1. 能力値推定の観点から

問題数を増やすことで推定誤差が減少したり、受験者数を増やして推定精度を向上することができると思われる。

2. インタフェースの観点から

能力値推定の結果を視覚的に分かりやすく表示する機能を追加することで、受験者が推定結果をより深く理解できるようにする。

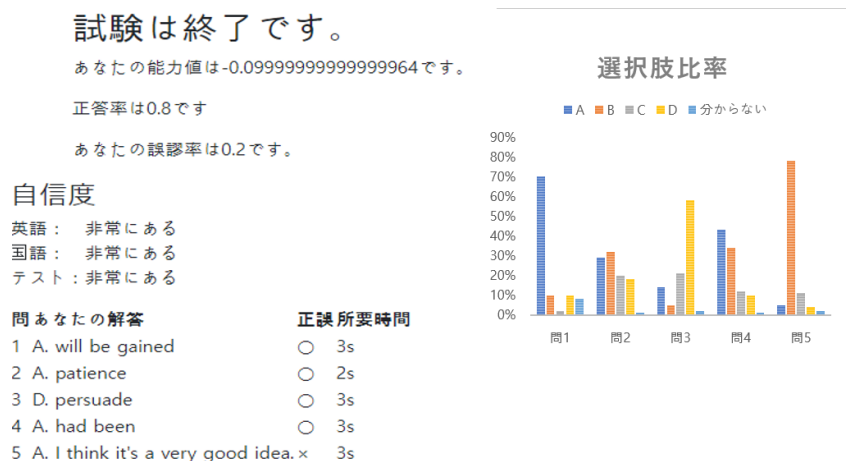


図 4: 追加改善案の例

例えば、自分の問いのレビューページにグラフやチャートを活用して、誤答の場合どの選択肢が正解であったが、各問題別の選択肢の選択される比率などを表示すると他の受験者と比較し自分の水準を把握できると思われる。また、テストが始まってから時間を測定してテスト中に見えるようにすることで、受験者にとって問題別の時間配分がしやすくなると思われる。

5 まとめ

今回の実験では HTML と JavaScript を基で現代テスト理論（項目反応理論）を適用したテストアプリケーションを作成した。アプリケーションのシステムの方、つまりバックエンドに触れながらその仕組みを理解し活用できるようになった。また、インタフェースでの面（フロントエンド）でもユーザの便利性的ために工夫しより使いやすいアプリケーションとは何かについて考察を行った。元々 Web 開発に興味があったので今回の実験を通じてよりその仕組みが理解できるようになった。

A 付録

Listing 1: 3_index.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
    bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/
      azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpUuCOmLASjC"
    crossorigin="anonymous">
</head>

<body>
  <form id="exam-box" class="container">
    <div id="question-area" class="bg-light p-4 m-2 rounded border"></div>
    <div id="choice-area"></div>
    <input class="btn btn-primary m-2" type="submit" value="解
      答" id="submit-button">
  </form>
  <p id="confidence-level" style="display: none;"></p>
  <table id="result-table" style="display: none;">
    <thead>
      <tr>
        <th class="tg-fymr">問</th>
        <th class="tg-7btt">あなたの解答</th>
        <th class="tg-7btt">正誤</th>
        <th class="tg-1wig">所要時間</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td class="tg-fymr">1</td>
        <td class="tg-dvpl" id="1_1"></td>
        <td class="tg-dvpl" id="1_2"></td>
        <td class="tg-dvpl" id="1_3"></td>
      </tr>
      <tr>
        <td class="tg-fymr">2</td>
        <td class="tg-dvpl" id="2_1"></td>
        <td class="tg-dvpl" id="2_2"></td>
        <td class="tg-dvpl" id="2_3"></td>
      </tr>
      <tr>
        <td class="tg-1wig">3</td>
        <td class="tg-0lax" id="3_1"></td>
        <td class="tg-0lax" id="3_2"></td>
```



```

        <td class="tg-01ax" id="3_3"></td>
    </tr>
    <tr>
        <td class="tg-1wig">4</td>
        <td class="tg-01ax" id="4_1"></td>
        <td class="tg-01ax" id="4_2"></td>
        <td class="tg-01ax" id="4_3"></td>
    </tr>
    <tr>
        <td class="tg-1wig">5</td>
        <td class="tg-01ax" id="5_1"></td>
        <td class="tg-01ax" id="5_2"></td>
        <td class="tg-01ax" id="5_3"></td>
    </tr>
</tbody>
</table>
<script src="1_functions.js"></script>
<script src="3_functions.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+
NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="
anonymous">
</script>
</script>
</body>

</html>

```

```

1 //   から 要素 (HTMLdivid="result) を取得"
2 // const result = document.getElementById('result');
3
4 // result.innerHTML = 'Result Div'
5 //演習 1-0
6 const practice0 = () => {
7     const min = -3;
8     const max = 3;
9     const step = 1;
10    for (let theta = min; theta <= max; theta += step) {
11        result.innerHTML += theta + ', ';
12    }
13 }
14 // practice0();
15
16 //演習 1-1
17 const norm = (theta) => {
18     return Math.exp(-theta * theta / 2) / Math.sqrt(2 * Math.PI);

```

```

19 }
20 const normDist = (min, max, step) => {
21   const dist = [];
22   for (let theta = min; theta <= max; theta += step)
23     dist.push(norm(theta) * step);
24   return dist;
25 }
26 // console.log(normDist(-3, 3, 1));
27 // const practice1 = () => {
28 //   const dist = normDist(-3, 3, 1);
29 //   for (let i = 0; i < dist.length; i++) {
30 //     result.innerHTML += dist[i].toFixed(3) + ', ';
31 //   }
32 // }
33 //practice1();
34
35 const practice1 = () => {
36   normDist(-3, 3, 1).forEach(value => {
37     result.innerHTML += value.toFixed(4) + ', ';
38   })
39 }
40 // practice1();
41
42 //演習 1-2
43 const correctProbability = (theta, a, b) => {
44   return 1 / (1 + Math.exp(-1.7 * a * (theta - b)));
45 }
46 const responseProbability = (x, theta, a, b) => {
47   const p = correctProbability(theta, a, b);
48   return Math.pow(p, x) * Math.pow(1 - p, 1 - x);
49 }
50 }
51 const icc = (x, a, b, min, max, step) => {
52   const iccDist = [];
53   for (let theta = min; theta <= max; theta += step) {
54     iccDist.push(responseProbability(x, theta, a, b));
55   }
56   return iccDist;
57 }
58 const practice2 = () => {
59   result.innerHTML = responseProbability(0, 1, 1, 2).toFixed(3) + '<br>';
60   icc(1, 1, 0, -3, 3, 1).forEach(value => {
61     result.innerHTML += value.toFixed(3) + ', ';

```

```

62   })
63 }
64 // practice2();
65
66 //演習 1-3
67 const itemBank = [
68   { a: 1, b: 0 },
69   { a: 0.5, b: 0.3 }
70 ];
71
72 const bayes = (x, itemBank, min, max, step) => {
73   const dist = normDist(min, max, step); //P θ ()
74   x.forEach((eachX, index) => {
75     const item = itemBank[index]; // から一つの問題セットを取得$itemBank
76     const likelihoodDist = icc(eachX, item.a, item.b, min, max, step);
77     dist.forEach((_, theta, arr) =>
78       arr[theta] *= likelihoodDist[theta]); //P θ Π ()P(X θ |)
79   });
80   //周辺尤度の計算
81   const marginalLikelihood = dist.reduce((a, b) => a + b); //P θ ()
82   dist.forEach((_, theta, arr) =>
83     arr[theta] /= marginalLikelihood); //P θ Π ()P(X θ |)
84   return dist;
85 }
86
87 const argmax = arr => arr.indexOf(arr.reduce((a, b) => Math.max(a, b)));
88
89 const estimation = (x, itemBank, min, max, step) => {
90   const probability = bayes(x, itemBank, min, max, step);
91   return min + argmax(probability) * step;
92 }
93
94 const practice3A = () => {
95   bayes([1], [{ a: 1, b: 0 }], -2, 2, 1).forEach(value => {
96     result.innerHTML += value.toFixed(3) + ', ';
97   })
98 }
99 const practice3B = () => {
100   result.innerHTML = estimation([1], [{ a: 1, b: 0 }], -2, 2, 1);
101 }
102 // practice3A();
103 // practice3B();
104

```

```

105 //演習 1-4
106 const information = (theta, itemBank) => {
107   let info = 0;
108   itemBank.forEach(item => {
109     const p = correctProbability(theta, item.a, item.b);
110     info += 1.7 * 1.7 * item.a * item.a * p * (1 - p);
111   });
112   return info;
113 }
114
115 const standardError = (theta, itemBank) => {
116   return 1.0 / Math.sqrt(information(theta, itemBank));
117 }
118
119 const practice4 = () => {
120   result.innerHTML = information(0, [{ a: 1, b: 0 }]).toFixed(4);
121 }
122 // practice4();
123
124 const cauchy = (x0, eta) => {
125   return x0 + eta * Math.tan(Math.PI * (Math.random() - 0.5));
126 }
127 /**
128  * シミュレーション実験
129  * @param {number} Q_NUM 問題数
130  * @param {number} E_NUM 受験者数
131  * @return {number} 平均誤差
132  */
133 const simulation = (Q_NUM, E_NUM, eta) => {
134   //アイテムバンク生成
135   const itemBank = [];
136   for (let i = 0; i < Q_NUM; i++) {
137     itemBank.push({
138       a: Math.random() * 2, //[0,2)
139       b: (Math.random() - 0.5) * 6 //[-3,3)
140     });
141   }
142
143   //受験者生成
144   const examinee = [];
145   for (let e = 0; e < E_NUM; e++) {
146     examinee.push((Math.random() - 0.5) * 6); //[-3,3)
147   }

```

```

148
149 //受験者生成コーシー分布 ()
150 const examinee2 = [];
151 for (let e = 0; e < E_NUM; e++) {
152     examinee2.push(cauchy(0, eta));
153 }
154 //受験者ごとの誤差
155 const error = [];
156 for (const theta of examinee2) {
157     const x = []; //正誤
158     for (const item of itemBank) {
159         x.push(correctProbability(theta, item.a, item.b) > Math.random() ? 1 : 0);
160     }
161     error.push(Math.abs(theta - estimation(x, itemBank, -3, 3, 0.1)));
162 }
163
164 //平均誤差
165 return error.reduce((a, b) => a + b) / E_NUM;
166 }
167
168 // result.innerHTML = 平均誤差: ' + simulation(50, 30, 1);

```

Listing 2: 1.functions.js

```

1 const getItem = async (i) => {
2     const response = await fetch('./2_itemBank.json?v=' + Date.now());
3     const data = await response.json();
4     return data[i];
5 }
6 //practice2-1
7 // const practice1 = () => {
8 //     getItem(5).then(item => console.log(item.choices[0]));
9 // }
10 // practice1();
11 const exam = {};
12 let questionNum = 1; //問題番号
13 const myChoice = []; //選択した解答
14 const ox = []; //正誤○×,
15 const questionTimes = []; //解答にかかった時間の配列
16
17 const createExam = (item) => { /* 問題文・選択肢の生成 */
18     startTime = Math.floor(new Date().getTime() / 1000); //時間測定の始まり
19     //問題文領域
20     const questionArea = document.getElementById('question-area');

```

```

21 questionArea.innerHTML = questionNum + '.' + item.question;
22
23 //選択肢領域
24 const choices = item.choices;
25 const choiceArea = document.getElementById('choice-area');
26 choiceArea.classList.add("m-3");
27 choiceArea.innerHTML = '';
28 choices.forEach((eachChoice, index) => {
29     //ラジオボタン
30     const input = document.createElement('input');
31     input.id = 'choice' + index;
32     input.type = 'radio';
33     input.name = 'choices';
34     input.value = index;
35     input.style.cursor = 'pointer';
36     input.classList.add("form-check-label");
37     input.required = true; //ラジオボタンが押されないと解答ボタンが押されない機能
38
39     //選択肢ラベル
40     const label = document.createElement('label');
41     label.setAttribute('for', 'choice' + index);
42     label.innerHTML = '&nbsp;&nbsp;&nbsp;' + eachChoice;
43     label.style.cursor = 'pointer';
44     label.classList.add("form-check-label");
45     //選択肢領域への追加
46     const div = document.createElement('div');
47     div.appendChild(input);
48     div.appendChild(label);
49     div.classList.add("form-check");
50     choiceArea.appendChild(div);
51 });
52 questionNum++;
53 const submitButton = document.getElementById('submit-button');
54 submitButton.addEventListener('click', () => { //解答ボタンが押されたとき実行さ
55     const endTime = Math.floor(new Date().getTime() / 1000); //解答が終わる時間
56     elapsedTime = endTime - startTime; //解答所要時間の計算
57 }
58 );
59 questionTimes.push(elapsedTime);
60 }
61 const startTesting = async () => { /* テスト開始時の処理 */
62     exam.n = 0; //解答数

```

```

63 exam.x = []; //正誤
64 exam.theta = 0; //能力値
65 exam.bank = []; //項目履歴 追加/**/
66 errorRate = 0; //誤謬率
67 createExam(await getItem(0));
68 }
69 startTesting();
70
71 // フォームが送信されたとき解答ボタンが押されたときの処理 ()
72 document.getElementById('exam-box').onsubmit = (e) => {
73   e.preventDefault(); //フォームの送信ページ更新の停止 ()
74   continueTesting(); //テスト継続処理の呼び出し
75 }
76
77 const continueTesting = async () => { /*テスト継続時の処理*/
78   //選択されているラジオボタンの値を取得
79   const choice = parseInt(document.getElementById('exam-box').choices.value);
80   //現在の問題項目を取得
81   let item = await getItem(exam.n);
82   //選択した解答の記録
83   myChoice.push(item.choices[choice]);
84   if (typeof item.correct !== 'undefined') {
85     //問題項目の保存
86     exam.bank.push(item);
87     //正答なら、誤答なら 1 0 を記録
88     exam.x.push(choice == item.correct ? 1 : 0);
89     //配列に正答なら ox ○、誤答なら × を記録
90     ox.push(choice == item.correct ? '○' : '×');
91     //能力値正答率を計算 ()
92     exam.theta = estimation(exam.x, exam.bank, -3, 3, 0.1);
93   }
94   //解答数を繰り上げ
95   exam.n++;
96   //次の問題項目を取得
97   item = await getItem(exam.n);
98   //次の問題があれば出力、なければテスト終了
99   if (typeof item !== 'undefined')
100     createExam(item);
101   else {
102     finishTesting();
103     const confidenceLevel = document.getElementById('confidence-level');
104     const englishConf = myChoice[0]; //英語自信度
105     const japaneseConf = myChoice[1]; //国語自信度

```

```

106     const testConf = myChoice[exam.n - 1]; //テスト全体
107     confidenceLevel.style.display = 'block';
108     confidenceLevel.innerHTML = '<h3自信度></h3>' + '英語： ' + englishConf + '<br>' + '
国語： ' + japaneseConf + '<br>' + 'テスト：' + testConf;
109     for (let i = 1; i <= 5; i++) {
110         const seigo = document.getElementById(i + '_2'); //正誤
111         seigo.innerText = ox[i - 1];
112         const t = document.getElementById(i + '_3'); //時間
113         t.innerText = questionTimes[i + 1] + 's';
114     }
115 }
116 };
117
118 const finishTesting = async () => { /*テスト終了時の処理 */
119     errorRate = (1 - exam.x.reduce((a, b) => a + b) / (exam.n - 3)).toFixed(1);
120     let result = '<h1試験は終了です。></h1>'
121         + '<pあなたの能力値は>' + exam.theta + 'です。</p>';
122     result += '<p正答率は>' + exam.x.reduce((a, b) => a + b) / (exam.n - 3) + 'で
123 す</p>' + '<pあなたの誤謬率は>' + errorRate + 'です。</p>';
124     document.getElementById('exam-box').innerHTML = result;
125     const resultTable = document.getElementById('result-table');
126     resultTable.style.display = 'table';
127     for (let i = 1; i <= 5; i++) {
128         const choice = document.getElementById(i + '_1');
129         choice.innerText = myChoice[i + 1];
130     }
131 }

```

Listing 3: 3_functions.js