

基礎化学教育のための web 上の学修支援の展開

中 川 邦 明

Extension of a Web-based Learning System
to Assist Students in Basic Chemistry Courses

Kuniaki NAKAGAWA

2014 年 11 月 21 日受理

要 旨

既報の web 上の自習テストシステムを、新たな内容、新たな科目へと拡張し、基礎化学分野での学修を多角的に支援できるようにした。

Abstract

The reported web-based training system was extended to a new subject matter and a new course, to assist students in basic chemistry courses.

1. 新たな学修支援の必要性

これまで、初等教育課程共通科目の理科 IA、IB、理科専攻科目の化学 I（旧基礎化学 I）について、web 上の自習テストシステムを開発・活用してきた¹⁻⁴⁾。これらの自習システムを活用して授業を進め、周期表や元素についての学生の理解は進んだが、学んだ元素が身の回りの具体的な物質とどう繋がっているかの理解は十分ではなかった。そこでこのような内容を自習システムに含めることが必要であると考えた。

また、理科専攻 3 年生に対する実験科目である化学実験については、これまで実験前に助手が予習状況を確認し、また実験終了時には測定の計算処理結果をパソコンによって学生自身に確認させてきた。しかしながら学生の数の増加と基礎力の低下の両面から、教育効果を維持するためには何らかの対策が必要となっていた。

以上のような観点から、これまでの web 自習システムを基礎に、より広範に学生の学修を支援するシステムへの展開が必要となった。

1.1 理科 I A における課題と対応の構想

理科 IA、IB、の自習テストにおいては、まず元素名、元素記号、周期表の記憶を定着させ、続いて基本的なイオンの符号と価数を（典型元素については周期表の族と関連させて）系統立てて記憶させ、さらに記憶した陰陽両イオンを価数の逆比で組み合わせることで電解質の化学式をつくる、という積み重ねを重視してきた。特に化学式については、様々なイオンをランダムに組み合わせることで化合物を出題することで、限られた化合物の化学式を個別に記憶することでは対応できない状況をつくり、上記の系統性を踏まえた理解が徹底するように企図してきた¹⁾。

この自習システムにおいて上記の意図はある程度達成されてきた²⁾が、一方で学生諸君が元素名と元素記号を単なる記号的処理の対象として扱い、自然界、日常生活で目に触れる具体的な物質とのつながりが見失われている傾向が感じられた。そのような観点から、理科 IA の最初の段階で、各元素と具体的な物質とのつながりについて、知識を定着させる問題を導入しようと考えた。問題のタイプとしては、

- (a) 説明文を提示し、それが説明している元素名と元素記号を解答させる。
- (b) 元素名一つと複数の説明文を提示し、当該元素に対応する説明文を選択させる。

とし、(b)のタイプを繰り返して記憶を定着させ、最後に(a)のタイプで確認するという段階を考えた。

1.2 化学実験における課題と対応の構想

理科専攻 3 年生を対象とした化学実験は、前期木曜午後 3、4 時限 2 コマ連続で開設され、中学校、高等学校理科免許取得のための実験科目である。受講の前提として、化学 I から始まる化学分野の理科専攻科目の講義・演習科目の系統的な理解と、事前のきちんとした予習とが要求されている。それらを満たせば所定時間内に終了できるように内容を精選し、実験操作の手順を細かく段階に分けて指示した実験プリントを用意しているが、多くの学生が 4 時限終了時には終わらないのが現実である。

このような状況への対応としてこれまで、「予習テスト」を用意し、それを前日までに解答させて明石助手がチェックしてきた。「予習テスト」の内容は、実験の前提となる現象・概念等の理解、実験操作上留意すべき点などに加えて、定量的な測定結果の得られる実験では、その測定結果をどう処理し、最終的に得たい量を求めるかのデータ処理手順がある。そのために模擬測定データを与えて計算し結果を求めさせる設問も「予習テスト」に含めてきた。ただ全員に同じ模擬データを与えると、中味を理解せずに結果のみ丸写しすることになるので、学生ごとに学籍番号をもとに異なる値となるような模擬データ生成手順を与え、明石

助手により事前に個別に確認してきた。しかし当然予想されるように多くの学生は期限ギリギリに提出してくるので、助手によるチェック確認業務の飽和状態を引き起こすこととなってきた。これに対して、web 自習テストタイプの「予習テスト」とすれば、期限間際に集中する学生の需要にも同時並行処理により対応して学修効果を高め、併せて担当教員の負担を軽減できる。

さらにこの定量的な測定結果の処理に対してこれまで、学生が実験終了時に各自の結果に基づいて誤りなく最終結果を計算できたかも確認できるようにしてきた。化学実験用のノートパソコンに Excel シートを用意し、各学生はそこに測定値を入力して計算し、表示された結果と各自のノートの計算結果が一致するかを確認する。学生の待ち行列を減らそうとパソコンの台数を増やすと、事後に複数のパソコンに入力されたデータを統合する手間がかかるばかりでなく、何段階かある実験では、最初の段階の結果を入れたパソコンに次の段階でも入力する必要があるために、他のパソコンが空いているにもかかわらず待ち行列ができたりする。スマートフォンの普及により、web 上に計算確認のための入出力のページを用意しておけば同時並行処理が可能となり、学生が各自のスマートフォンで web を介して随時確認でき、またその結果もすべてサーバーに一括して保存・管理ができることになる。

2. 支援システムの構築 — JavaScript によるモジュール作成の実際

前節で述べた構想を具体的にどのように実現したかについて、個別のモジュールをどのように配列してシステムを構成したかをまず概観し、次いで各モジュールの中から代表的なものいくつかを例として、JavaScript によるコーディングの工夫点等についてソースリストの一部をあげて具体的に述べることとする。

2.1 理科 IA の学修支援の展開

これまでの理科 I 自習テストの目次では筆頭に「1. 元素記号と元素名」が置かれていたが、その前に「0. 元素を知ろう」を置き、対象元素の範囲を、「0.1 第 2 周期まで」、「0.2 原子番号 11 ~ 29 のうち記憶すべきもの」、「0.3 原子番号 30 ~ のうち記憶すべきもの」、「0.4 まとめテスト (記憶すべき元素すべて)」、の 4 段階に分けて 1.1 に述べた(b)タイプのテストによって段階的に学べるようにした。これらを学び終えた後、(a)タイプのテスト、「0.5 元素の説明→元素名、元素記号 (まとめ)」で復習をする。

2.1.1 元素を知ろう(a)タイプ (0.5 元素の説明→元素名、元素記号 (まとめ))

学修の順序からは後になるが、(a)タイプはアルゴリズム的には(b)タイプより簡単で、従来の例えば「1. 元素記号と元素名」と大局的には違いはないので最初

に説明する。その実行画面の例を図 1 に示す。これまでの画面に比べてボタンの数を少なくし、各場面に応じてボタンの機能が変化し、ボタンの表示も変化するようにした。ボタンの数が多いと、その段階で押すべきでないボタンを誤って押す等の不測の事態が発生するのを未然に防止するためである。

画面中央の説明文が表示されている場所は text00 と名付けたテキストエリアで、スタイル指定でボーダーラインなしとした。その下の右寄りに元素名と元素記号を入力するテキストボックス（それぞれ text01、text02、と名付けた）が配置され、そこに解答を入力した上で [判定] ボタン (text03、起動時には、[開始] ボタンとなっており、そこをクリックしてスタートする。) をクリックすると元素名と元素記号の下の小さなテキストボックス (それぞれ text04、text05) に正誤が○×で表示される。元素名と元素記号の両方が正解の場合には下の正解数欄 (text06 という名のボーダーラインなしのテキストボックス) の数値が 1 つ増え、その元素は改めて出題されることはない。どちらかが不正解の場合は、後ほど再出題される。また「判定」ボタンの表示は「次へ」に変わり、そこをクリックすると次の問題として新たな元素の説明文がテキストエリアに表示され、ボタンの表示は再び「判定」となり、次の元素名と元素記号の入力待ちとなる。

0.5 元素の説明→元素名、元素記号(まとめ)

このページは、01～04 のページで学んだことの最終的な確認テストとして行なうものです。
下の「開始」ボタンを押して始めて下さい。

説明文

食塩の構成元素の一つ。単体は軟らかい金属で、反応性が高く、空気中の酸素で酸化され、水と激しく反応して水素を発生する。そのため石油中に保存する。黄色の炎色反応を示し、照明に用いられるこの元素のランプも同じ色。水酸化物は苛性ソーダとも呼ばれ、強いアルカリ性で油酸から石鹸をつくるのに用いられる。

は、何という元素についての説明ですか？ 右の枠内に入力し、下の「判定」ボタンを押して下さい。元素名 (元素記号)

これまでのところ、38 問中、1 問できました。
不正解の問題は、あとからまた出てきます。全問正解できて終了です。

終了する時は、下のボタンを押して、結果を保存することを忘れないように。
「0. 元素を知ろう」の

図 1 理科 I テスト 0.5 の実行時画面

この処理の JavaScript のソースコードをリスト 1 に掲げた。shuffle(array)、timetest()、getCookie(key)、setCookie(key, val) の諸関数は、既に報告^{1,2)}したので再録しない。配列 eleSym、eleName、eleDesc は、元素記号、元素名、元素の説明をそれぞれ記憶するためのものである。eleDesc の内容は様々な元素の紹介^{5,6)}を参照して作成した膨大なものとなるのでリスト 1 では最初と最後の配列要素 (水素と鉛) のみを残し、他は割愛した。

プログラムの実行状態をあらわすフラグの一つとして、statflag を用いて

いる。その値は問題が提示されて解答入力・判定要求待ちの状態 (text03 ボタンの表示は [判定]) で 1、解答入力後、判定が表示され、次の問題の出題要求待ちの状態 (text03 ボタンの表示は [開始]、[次へ]、[再開] いずれかの状態) で 0 となっている。text03 ボタンが押されると、関数 contProb() が実行され、statflag が 1 であれば関数 checkAns() が、0 であれば関数 putProb() がそれぞれ実行される。putProb() が実行されると statflag は 1、text03 ボタンの表示は [判定] となり、checkAns() が実行されると statflag は 0、text03 ボタンの表示は [次へ] となる。

関数 valInit() は BODY タグの onload 属性で指定してあるので、HTML ファイル読み込み時に実行される。また、[中断] ボタンをクリックした時にも実行される。そのため、記録等の初期化はこの関数では行っていないことに注意する必要がある。

最後のデータの保存については、従来は

```
<form name="myform" action="cgi-bin/group01v2.cgi"
method="post"><input type="submit" value="「周期表のテスト」の目次に戻る "></form>
```

のようにいきなり cgi プログラムに制御を渡していたが、今回は

```
<form method="post">「0. 元素を知ろう」の <input type="button"
value=" 目次に戻る " onClick="saveData(this.form)"></form>
```

のように、一旦 JavaScript 関数 (ここでは saveData(f)) を実行し、その中から cgi プログラムに制御を渡すようにした⁷⁾。このことにより、解答の途中で突然中断した場合にも、その時点での経過時間と正解数を記録することが可能となる。突然の中断か否かを判断するためのフラグが recflg である。このフラグは text03 ボタン ([開始] または [再開]) ボタンによって putProb() が実行されるごとに 0 にリセットされ、すべての問題が終った時点で経過時間等を記録用の変数に追記し、フラグを 1 にセットする。[中断] ボタンが押され、abortProb() が実行された場合も、同様にそれまでの経過時間を追記してフラグを 1 にセットする。そのような過程を経ず、いきなり [目次に戻る] ボタンが押されて saveData(f) が実行された場合には、recflg フラグが 1 になっていないので、改めてその時点での情報を記録用変数に追記し、recCookie() によりクッキーに記録してから cgi プログラムに制御を渡し、サーバー上に記録を書き込む。

リスト 1

```
var eleSym = new Array(
"H", "He", "Li", "Be", "B", "C", "N", "O", "F", "Ne",
```

```

"Na", "Mg", "Al", "Si", "P", "S", "Cl", "Ar", "K", "Ca", "Cr", "Mn",
"Fe", "Co", "Ni", "Cu", "Zn", "Ge", "As", "Br", "Ag", "Cd", "Sn", "I",
"Ba", "Au", "Hg", "Pb");

var eleName = new Array(
"水素", "ヘリウム", "リチウム", "ベリリウム", "硼素", "炭素",
"窒素", "酸素", "フッ素", "ネオン", "ナトリウム", "マグネシウム",
"アルミニウム", "珪素", "燐", "硫黄", "塩素", "アルゴン", "カリウ
ム", "カルシウム", "クロム", "マンガン", "鉄", "コバルト", "ニッケ
ル", "銅", "亜鉛", "ゲルマニウム", "砒素", "臭素", "銀", "カドミ
ウム", "錫", "ヨウ素", "バリウム", "金", "水銀", "鉛");

var eleDesc = new Array(
"単体は無色、無臭、可燃性の軽い気体で、空気との混合物は爆発する。
水の構成元素の一つ。原子核は元素中最も簡単で、宇宙に最も多く存在し、
太陽をはじめとする恒星はその核融合反応によって光っている。",
【中略】
"単体は古くから知られた融点の低い軟らかく重い金属で、水道管、銃弾、
放射線、X線の遮蔽材料、蓄電池の電極などに用いられる。錫との合金は
ハンダ、アンチモンとの合金は活字に用いられる。この元素を含む有機
化合物が自動車ガソリンの品質改良剤として添加されていたが今は禁止。"
);

var eleNo = new Array;
var timerc = "";
var record = "";
var countrc = "";
date = new Date();

function valInit() {
    tt = 0;
    count = 0;
    eleNo = new Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
        16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
        29, 30, 31, 32, 33, 34, 35, 36, 37);

    statflag = 0;

```

```
t0 = timetest();
document.myForm0.text06.value = count;
}

function putProb() {
  recflg = 0;
  if(eleNo.length > 0) {
    shuffle(eleNo);
    id0 = eleNo[0];
    document.myForm0.text00.value = eleDesc[id0];
    document.myForm0.text01.value = "";
    document.myForm0.text02.value = "";
    document.myForm0.text04.value = "";
    document.myForm0.text05.value = "";
    statflag = 1;
    document.myForm0.text03.value = "判定";
  }
}

function contProb(){
  if(statflag == 1){
    checkAns();
  }else{
    putProb();
  }
}

function abortProb(){
  record += "|";
  tt = timetest()-t0;
  date.setTime(tt);
  alert("中断しました。この問題を最初からやり直すには " +
        "「再開」を、やめて戻る時はページの一番下の " +
        "「目次に戻る」をクリックして下さい。");
  timerc += date.getMinutes()+":"+date.getSeconds()+"|";
  countrc += String(count)+"|";
```

```
document.myForm0.text03.value = "再開";
recflg = 1;
valInit();
}

function checkAns(){
  statflag = 0;
  ifcorr = 1;
  document.myForm0.text03.value = "次へ";
  eleN = document.myForm0.text01.value;
  eleS = document.myForm0.text02.value;
  if(eleN == eleName[id0]){
    document.myForm0.text04.value = "o";
  }else{
    ifcorr = 0;
    document.myForm0.text04.value = "x";
    record += eleName[id0] + "_" + eleN + ",";
  }
  if(eleS == eleSym[id0]){
    document.myForm0.text05.value = "o";
  }else{
    ifcorr = 0;
    document.myForm0.text05.value = "x";
    record += eleSym[id0] + "_" + eleS + ",";
  }
  if(ifcorr == 1){
    eleNo.shift();
    count++;
  }
  document.myForm0.text06.value = count;
  if(count >= 38){
    record += "|";
    tt = timetest()-t0;
    date.setTime(tt);
    alert("お疲れ様でした。これで終わりです。
      " + date.getMinutes() + "分" + date.getSeconds())
```



```

        + "秒 かかりました。¥n" + "この問題をもう一度最初からや
        る" + "には「再開」を、やめて戻る時はページの一番下の" +
        "「目次に戻る」をクリックして下さい。");
    timerc += date.getMinutes()+":"+date.getSeconds()+"|";
    countrc += String(count)+"|";
    document.myForm0.text03.value = "再開";
    recflg = 1;
    valInit();
    }
}

function saveData(f){
    if(recflg != 1){
        countrc += String(count)+"|";
        record += "|";
        alert(record);
        date.setTime(tt);
        timerc += date.getMinutes()+":"+date.getSeconds()+"|";
    }
    recCookie();
    f.action = "cgi-bin/ele05.cgi";
    f.submit();
}

function recCookie(){
    uid = getCookie('uid');
    sct = setCookie('time', timerc);
    sct = setCookie('count', countrc);
    scr = setCookie('recd', record);
}

```

2.1.2 元素を知ろう(b)タイプ (0.1 元素を知ろう (第2周期まで))

(b)タイプの具体例として、最も簡単な0.1元素を知ろう(第2周期まで)を例として説明する。その実行画面の例を図2に示す。[判定]、[中断]などのボタンの機能は上記(a)タイプと同じである。

0.1 元素を知ろう(第2周期まで)

下の「開始」ボタンを押して始めて下さい。

炭素(元素記号 c)についての最も適当と思う説明の上のラジオボタンをクリックし、下の「判定」ボタンを押して下さい。

単体は無色、無臭の気体で、空気中に約78%含まれるが、反応性が小さいため目立たない。アミノ酸、タンパク質、DNA等の生命維持に欠かせない化合物の構成要素であり、肥料の3要素の一つとしても不可欠。液体は冷却材として低温の実験に用いられる。

エメラルドやアクアマリンなどの仲間である緑柱石(ベリル)から得られ、それに因んだ名前。単体は堅い金属。X線を透過するため、X線機器の窓に用いられる。

生物体をつくる有機化合物の中心となる元素。単体はダイヤモンド、黒鉛、フラーレン等の同素体として知られる。

単体は無色、無臭、可燃性の軽い気体で、空気との混合物は爆発する。水の構成元素の一つ。原子核は元素中最も簡単で、宇宙に最も多く存在し、太陽をはじめとする恒星はその核融合反応によって光っている。

「判定」を押して判定結果が出てから、「次へ」を押すまでの時間はカウントされませんので、ゆっくり確認し、ノートに記録し、記憶してから「次へ」ボタンを押して次に進んで下さい。

判定

これまでのところ、10問中、問できました。
不正解の問題は、あとからまた出てきます。全問正解できて終了です。

中断

終了する時は、下のボタンを押して、結果を保存することを忘れぬように。

「0. 元素を知ろう」の [目次に戻る](#)

図2 理科 I テスト 0.1 の実行時画面

元素名と元素記号が提示される(上の例では炭素)ので、その元素に関する説明を、表示された4つの中からラジオボタンで選び「判定」をクリックする。4つのラジオボタンは form 中の第2~5要素、その下の4つの説明文は第6~9要素、さらにその下に、「判定」後に正解を示すためのテキスト要素が4つ、第10~13要素として JavaScript によって参照される。

ここで毎回異なる順序で元素を選択して出題することは、これまでの自習テストと同様¹⁾であるが、4つの選択枝のうち不正解としてどの3つの元素の説明文を出すかと、4つの説明文を並べる順序についての工夫が新たに必要である。そのためのアルゴリズムの中心となる部分をリスト2に示した。

リスト2

```
function valInit() {
    tt = 0;
    count = 0;
    eleNo = new Array(0,1,2,3,4,5,6,7,8,9);
    eleNo2 = new Array(0,1,2,3,4,5,6,7,8,9);
    statflag = 0;
}
```

```
function putProb() {
  recflg = 0;
  if(eleNo.length > 0) {
    shuffle(eleNo);
    id0 = eleNo[0];
    document.myForm0.text01.value = eleName[id0];
    document.myForm0.text02.value = eleSym[id0];
    shuffle(eleNo2);
    for (i=0 ; i<eleNo2.length ; i++){
      if (eleNo2[i] == id0){
        eleNo2[i] = eleNo2[0];
        eleNo2[0] = id0;
        break;
      }
    }
    jkey = Math.floor(4*Math.random());
    if(jkey != 0){
      eleNo2[0] = eleNo2[jkey];
      eleNo2[jkey] = id0;
    }
    for (i=0 ; i<4 ; i++){
      document.myForm0.elements[i+2].checked = false;
      document.myForm0.elements[i+6].value =
        eleDesc[eleNo2[i]];
      document.myForm0.elements[i+10].value = "";
    }
    t0=timetest();
    statflag = 1;
    document.myForm0.elements[14].value = "判定";
  }
}

function checkAns(){
  tt += timetest()-t0;
  statflag = 0;
  document.myForm0.elements[14].value = "次へ";
}
```

```

if(document.myForm0.elements[jkey+2].checked){
    document.myForm0.elements[jkey+10].value = " 正解です ";
    count++;
    eleNo.shift();
}else{
    document.myForm0.elements[jkey+10].value =" こちらが正
解 ";
    for (i=0 ; i<4 ; i++){
        if(document.myForm0.elements[i+2].checked){
            errSym = eleSym[eleNo2[i]];
            break;
        }
        errSym = "";
    }
    record += eleSym[id0] +"_" + errSym +", ";
}
document.myForm0.text03.value =  count;
if(count >= eleNo2.length){
    record += "|";
    date.setTime(tt);
    alert("お疲れ様でした。これで終わりです。" + date.
getMinutes() + "分" + date.getSeconds() + "秒 がか
りました。¥n" + "この問題をもう一度最初からやるには「再開」を、
やめて戻る時は " + " ページの一番下の「目次に戻る」をクリック
して下さい。");
    timerc += date.getMinutes()+" ":" +date.
getSeconds()+"|";
    countrc += String(count)+"|";
    document.myForm0.elements[14].value = " 再開 ";
    recflg = 1;
    valInit();
}
}
}

```

出題する元素と元素記号は、従来と同様の手順¹⁾で配列 eleNo を関数 shuffle() によって並べ替え、先頭の要素を id0 に記憶し、それをキーとし

て配列 eleName と eleSym を参照する。正解の場合は関数 checkAns () 内で shift () メソッドによって先頭の要素を取り除いて 1 つずつ詰めて次の問題に進む。

他方の配列 eleNo2 は 4 つの選択肢の並び順を決めるために導入したもので、毎回同様に並べ替えて先頭から 4 つの要素を取るのであるが、選択肢の中に正答が入っていないといけなないので、id0 を先頭の要素と入れ替える。これではいつも先頭が正答となってしまうので、さらに 0 ~ 3 の 4 つの整数のどれかを乱数により発生させ jkey に記憶し、正解キーを記憶した先頭の配列要素 (第 0 要素) を第 jkey 要素と入れ替え、正解の位置が毎回変わるようにする。

2.2 化学実験の web による学修支援

化学実験については、これまで理科 I、化学 I 等で設けたと同様の入口ページを作成 (URL は <http://member.tokoha-u.ac.jp/~kuninaka/chemlab/index.htm>) し、履修者は学籍番号を入力すると図 3 のような目次のページが開く。なお、今回も半角の 0 を 6 個並べてゲストとして入ることができるので、実際に試行していただきたい。

以下では、計算手順にかかわる予習確認、計算のかかわる結果確認、についてそれぞれ具体例を挙げて説明し、最後に計算手順にかかわらない予習確認について述べたい。

2.2.1 化学実験の予習確認 (計算手順にかかわるもの)

計算手順に関する予習確認の具体例として、第 5 回の中和滴定の実験の一つをとりあげる。これ以外の計算手順にかかわる予習確認のモジュールも計算に関する部分はそれぞれ異なるが、大筋のアルゴリズムは同様である。

中和滴定の実験の最初は、酸標準溶液の調製である。蓚酸二水和物を秤量瓶で精秤して純水に溶かし、メスフラスコで定容する。空の秤量瓶の質量と蓚酸二水和物を入れた秤量

第 1 回	クロマトグラフィー		予習確認	—
	食塩の溶解度と飽和食塩水の密度	実験手順	予習確認	—
		計算処理	予習確認	結果確認
	プラスチックの密度		予習確認	—
	空気の状態と密度		予習確認	—
第 2 回	食塩水の濃度と密度	実験手順	予習確認	—
		計算処理	予習確認	結果確認
	炎・ビュリ缶!?		予習確認	—
第 5 回	酸標準溶液の調製	実験手順	予習確認	結果確認
		計算処理	予習確認	
	水酸化ナトリウム水溶液の標定	実験手順	予習確認	
		計算処理	予習確認	
塩酸・酢酸の標定		予習確認		
第 6 回	滴定曲線		予習確認	—
第 8 回	物質の色とスペクトル	波長と色	予習確認	—
		透過率と吸光度	予習確認	
第 9 回	澱粉と糖		予習確認	—
第 10 回	澱粉の加水分解		予習確認	—
第 11 回	色素合成		予習確認	—

お気づきの点は中川までお知らせ下さい。

図 3 化学実験学修支援システムの目次画面

瓶の質量の 2 つの測定値を用いて標準溶液の濃度を求め、これを基準にして他の酸および塩基の水溶液の濃度を滴定により決定する。この標準溶液の濃度計算が正しくできるかの確認である。

実行画面を図 4 に示す。画面に表示された質量の模擬データ 2 つをもとに、各自で濃度を計算し、次の行の入力フィールドに入力し、[解答確認] をクリックする。プログラムは入力された濃度が設定した計算誤差の範囲に入っているかどうかの判定結果を表示する。

第5回 中和と中和滴定

酸標準溶液の調製

予習の確認 結果処理について

実験操作手順に従って測定し、空の秤量瓶は 15.152 g、蓚酸を入れて秤量すると、16.710 g、であったとする。

このとき、手順に従って調製された蓚酸標準溶液の物質濃度は有効数字4桁で、 mol/L となる。
原子量は、有効数字を考えて、必要な桁まできちんと調べて用いよ。

解答確認

「化学実験 予習と結果の確認」の [目次に戻る](#)

図 4 酸標準溶液の調製の予習問題の実行時画面

JavaScript によるプログラムのリストをリスト 3 に示す。関数 valInit() においては、まず秤量瓶と蓚酸二水和物の質量の模擬データをそれぞれ 15 ~ 16 g、1.55 ~ 1.65 g の範囲で Math クラスの random() 関数によって生成する。値は parseInt() 関数を用いて、いずれも 10^{-3} g の桁未満を切り捨ててそれぞれ、変数 WBottle と Acid に記憶する。空秤量瓶と蓚酸入秤量瓶の質量として、toFixed(3) メソッドを用いて小数第 3 位まで画面に表示する。またこれらの模擬データで計算した濃度は変数 Molar に記憶する。

画面に表示された模擬データで学生が濃度を計算し、入力フィールドに入力し、[解答確認] ボタンをクリックすると、関数 endProb() が実行され、入力値と Molar との相対誤差が 10^{-3} を超えたか否かで正誤を判定し、表示する。結果の保存等の処理は既報¹⁾ と基本的に同様であるので割愛する。

リスト 3

```
var correctAns = 0.0;
```

```
function valInit() {
  WBottle = 15 + parseInt(1000*Math.random())/1000;
  Acid = 1.55 + parseInt(100*Math.random())/1000;
  Molar = Acid*4/126.068;
  record = WBottle + "," + Acid + "," + Molar + "|";
  document.myForm0.elements[0].value = WBottle.toFixed(3);
  WBA = WBottle + Acid;
  document.myForm0.elements[1].value = WBA.toFixed(3);
  correctAns = Molar;
}

function endProb() {
  inderr = 0;
  inputAns = document.myForm0.elements[2].value;
  relerr = Math.abs((inputAns - correctAns)/correctAns);
  if(relerr < 0.001){
    document.myForm0.elements[2].style.backgroundColor =
      "pink";

    record += " ○ |";
  }
  else{
    document.myForm0.elements[2].style.backgroundColor =
      '#ffffff';

    record += inputAns + "|";
    inderr = 1;
  }
  if(inderr == 0){
    alert(" おめでとう。正解です！ 記録を残すため、必ず " +
      " ページの一番下の「化学実験 予習と結果の確認」の目次に戻る "
      +
      " ボタンをクリックして戻って下さい。");
  }
  else{
    alert(" 残念ですが、誤っています。もう一度、有効数字に " +
      " 注意して計算し直して、入力し直して下さい。");
  }
}
```

```

    }
}
    
```

2.2.2 化学実験の結果処理確認

実験結果処理確認の具体例として、やはり第5回の中和滴定の実験をとりあげる。中和滴定は、前項で予習確認として触れた「酸標準溶液の調製」に続いて、その標準溶液を用いて「水酸化ナトリウム水溶液の標定」を、標定した水酸化ナトリウム水溶液を用いて「塩酸・酢酸の標定」を行なう。各段階で得た濃度を次の段階で使用するのので、すべての実験が終了した段階でデータ処理と確認をまとめて行なうよりも、各段階ごとに濃度を決定してから次の段階に進むことが適切である。データの確認を各段階で行なう場合、各段階の測定にはそれなりの時間がかかるため、各段階ごとに起動し直して確認することとなる。その際、前の段階で決定した濃度をその都度入力しなおすのは面倒であるばかりか、間違いの原因

ともなりやすい。スマートフォンなど、各学生が自分の所有する機器を用いる場合が多いので、クッキーを利用して結果の濃度を受け渡すのが合理的であると考えた。図5の実行画面で、学生は実験で得たデータを入力し、[⇒計算⇒] ボタンをクリックし、現われた結果を各自の計算結果と比較する。線で区切られた段階が終了するごとに画面の説明に従って終了すると、クッキーにその段階で求めた濃度が保存される。そして次の段階のデータ確認のために再び起動した際にクッキーからその濃度が読み込まれて表示され、次の段階

第5回 中和中和滴定

実験結果の確認

酸標準溶液の調製

各自の測定結果を下の特内に入力し、「計算」をクリックし、出た結果と比べて各自の計算結果を確認しなさい。

空の秤量皿 g
 秤殻+秤量皿 g **⇒計算⇒** 調製された酢酸標準溶液の物質濃度 mol/L
 酢酸

このボタン をクリックし、ここまでの結果を保存して、一旦、終了して下さい。
 同じパソコンを使うと、ここまでの結果が保存されているので、以下の続きの処理が後からできます。
 (クッキーを保存する設定) しておいてください。

水酸化ナトリウム水溶液の標定

上の右側に既に計算した酸標準溶液濃度が表示されていることを確認して下さい。
 次に各自の測定結果を下表に入力し、「計算」をクリックし、出た結果と比べて各自の計算結果を確認しなさい。
 (平均に取込んだデータのみ入力、4回以降は空欄でも可)

体積/mL	第1回	第2回	第3回	第4回	第5回
滴定開始前	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴定終了後	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴下体積	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

滴下体積の平均 mL
 水酸化ナトリウム水溶液の物質濃度 mol/L

⇒計算⇒

このボタン をクリックし、ここまでの結果を保存して、一旦、終了して下さい。
 同じパソコンを使うと、ここまでの結果が保存されているので、以下の続きの処理が後からできます。
 (クッキーを保存する設定) しておいてください。

塩酸・酢酸の標定

塩酸

体積/mL	第1回	第2回	第3回	第4回	第5回
滴定開始前	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴定終了後	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴下体積	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

滴下体積の平均 mL
 塩酸の物質濃度 mol/L

⇒計算⇒

酢酸

体積/mL	第1回	第2回	第3回	第4回	第5回
滴定開始前	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴定終了後	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
滴下体積	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

滴下体積の平均 mL
 酢酸の物質濃度 mol/L

⇒計算⇒

このボタン をクリックし、以上の結果を保存して、終了して下さい。

図5 中和滴定の実験結果確認の実行画面

の計算にも用いられる。

以上の処理を、最初の 2 段階、即ち水酸化ナトリウム水溶液の標定までにかかわる部分のみを抜き出したリスト 4 で見てみよう。第 1 段階では関数 `calcData0()` で酢酸標準溶液の濃度を計算し、変数 `Molar0` に記憶する。さらに第 1 段階の終了手順で起動する関数 `saveData0(f)` によってクッキーに保存される。このクッキーは `uid` に記憶された学籍番号の末尾に文字列 `v0` を付け加えた名前（変数 `uidv0` に記憶）で参照される。同様に第 2 段階で求めた水酸化ナトリウム水溶液の濃度は関数 `calcData1()` によって変数 `Molar1` に記憶され、さらに関数 `saveData1(f)` によって、変数 `uidv1` に記憶した名前でクッキーに保存される。これらのクッキーに記憶された値は起動時に関数 `valInit()` によって読み込まれる。

上記以外のアルゴリズム等については、特に説明を要しないであろう。

リスト 4

```
function valInit() {
  uid = getCookie('uid');
  uidv0 = uid + 'v0';
  Molar0 = getCookie(uidv0);
  document.myForm00.elements[2].value=
      Math.round(10000*Molar0)/10000;
  uidv1 = uid + 'v1';
  Molar1 = getCookie(uidv1);
  document.myForm01.elements[17].value =
      Math.round(10000*Molar1)/10000;
}

function calcData0(){
  WBottle = document.myForm00.elements[0].value;
  WBAcid = document.myForm00.elements[3].value;
  Acid = WBAcid - WBottle;
  document.myForm00.elements[4].value =
      Math.round(1000*Acid)/1000;
  Molar0 = Acid*4/126.068;
  document.myForm00.elements[2].value =
      Math.round(100000*Molar0)/100000;
```

```
record0 =
WBottle + "," + WBAcid + "," + Acid + "," + Molar0 + "|";
}

function saveData0(f){
    setCookie('recd', record0);
    setCookie('uidv0', Molar0);
    f.action = "cgi-bin/chk5.cgi";
    f.submit();
}

function calcData1(){
    record1 = "";
    idata = 0;
    sum = 0;
    for (i = 0; i < 5; i++){
        if(document.myForm01.elements[i].value != "") {
            Volume = document.myForm01.elements[i+5].value
                    - document.myForm01.elements[i].value;
            document.myForm01.elements[i+10].value =
                Math.round(100*Volume)/100;

            sum += Volume;
            idata ++;
            record1 = record1 + "," + Volume;
        }
    }

    aveVolume = sum/idata;
    document.myForm01.elements[16].value =
        Math.round(100*aveVolume)/100;
    Molar1 = 20*Molar0/aveVolume;
    document.myForm01.elements[17].value =
        Math.round(100000*Molar1)/100000;
}

function saveData1(f){
    setCookie('recd', record1);
```

```

setCookie(uidvl, Molar1);
f.action = "cgi-bin/chk5.cgi";
f.submit();
}

```

2.2.3 化学実験の予習確認（計算手順にかかわらないもの）

化学実験支援モジュールのタイプとして、既に述べたもの以外は、実験の背景となる基本的な知識等の予習確認のためのものである。その多くは、旧来の自習テストの解答形式であるテキストボックスとセレクトボックスに加えて、ラジオボタンを使っているもの、web 上の動画等を参照するリンクを組み込んだものなどがあるが、ここでは出題に変化を持たせるために多少複雑なアルゴリズムを使用した、光の波長と色の関係の確認テストを紹介する。

実行画面を図 6 に示す。7つの波長に対して7つの色となっているので、長波長から順に、赤、橙、黄、緑、…と割り振っていくと正解できそうにも思えるが、そうならないように作った。リスト 5 に示すように、波長範囲を 8つ（赤が 2つ）に区切り、その中から 7つをランダムに選択している。色と波長の関係は参照する文献により少しずつ異なるので、異論の余地のない範囲を設定し、波長範囲の下限と幅を配列 lowWave と widWave に記憶した。従って、決して出題されない波長範囲が存在する。設定波長範囲内で乱数により出題する波長を決めている。「解答確認」ボタンをクリックすると、正解の色のラジオボタンをピンクで示すようにした。

第8回 物質の色とスペクトル

予習の確認 光の波長と色

次の波長の光の色として適当だと思うものを一つ選びなさい。解答の入力が終わったら下の「解答確認」ボタンを押さない。

- 412 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 605 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 530 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 478 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 688 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 582 nm 赤、 橙、 黄、 緑、 青、 藍、 紫
- 440 nm 赤、 橙、 黄、 緑、 青、 藍、 紫

解答確認

「化学実験 予習と結果の確認」の [目次に戻る](#)

図 6 物質の色とスペクトルの予習の実行時画面

リスト 5

```
var indAns = new Array(0,1,2,3,4,5,6,7);
var lowWave = new Array(680,630,595,580,520,470,435,380);
var widWave = new Array(50,50,15,10,30,20,15,40);

function valInit() {
  shuffle(indAns);
  shuffle(indAns);
  record = indAns + "|";
  for (i = 0; i < 7; i++){
    ida = indAns[i];
    Wavelen =
      lowWave[ida] + Math.round(widWave[ida]*Math.random());
    record += Wavelen + ",";
    document.myForm0.elements[8*i].value = Wavelen;
  }
}

function endProb() {
  count = 0;
  // 正解のラジオボタンに色をつける
  for (i=0 ; i<7 ; i++)
  {
    ida = indAns[i];
    if(ida > 0){ida = ida - 1;}
    document.myForm0.elements[8*i+ida+1].style.
      backgroundColor = "pink";
    if(document.myForm0.elements[8*i+ida+1].checked)
      count++;
  }
  // 全ラジオボタンの状態を記録
  for (ii=0 ; ii<56 ; ii++)
  {
    if(document.myForm0.elements[ii].checked)
```

```

        {record += "1";}else{record += "0";}
    }
    record += "|";
    if(count == 7)alert("おめでとう。全問正解です！ ¥n");
    alert("ピンクのついたところが正解です。正解を確認したら、" +
    "記録を残すため、必ずページの一番下の「化学実験 予習と結果の確認」"
    + "の目次に戻る ボタンをクリックして戻って下さい。");
}

```

3. まとめとこれからと

出題形式の工夫、使いやすい画面設計、変則的な中断の場合の記録の確保、さらに化学実験支援関係では、模擬データの発生、有効数字等に配慮した入出力書式の設定、丸め誤差に配慮した正誤判定、等のアルゴリズムを JavaScript を用いて実現することができた。また、入力フォームとして、従来のテキストボックス、セレクトボックスに加えて、ラジオボタンなど、多様な入力フォームを目的に合わせて活用できたことも意義深かった。授業を担当していく上で、必要に迫られて行ったシステムの拡張、展開であったが、所期の当該授業における必要性への対応を越えて、このような JavaScript のより多様な利用法を拓くことができた。

今後は、これまでに築きあげた学修支援システムを活用して学修支援を進めるとともに、学生の学修状況への目配りを欠かさず、改善と新たな必要性の機会をとらえては、蓄積したアルゴリズムと多様な利用法の土台の上に、さらに学修支援への試みを続けていきたい。

引用

- 1) 中川邦明、常葉学園大学研究紀要 教育学部 30, 351–376 (2010).
- 2) 中川邦明、常葉学園大学研究紀要 教育学部 31, 103–115 (2011).
- 3) 中川邦明、常葉学園大学研究紀要 教育学部 32, 1–8 (2012).
- 4) 中川邦明、常葉大学教育学部紀要 34, 267–278 (2014).
- 5) 一家に一枚 周期表 (http://www.mext.go.jp/a_menu/kagaku/week/1298266.htm)
- 6) セオドア・グレイ、若林文高 監訳、「世界で一番美しい元素図鑑」創元社 (2010).
- 7) <http://oshiete1.nifty.com/qa7144455.html> の記述を参考にした。