第1章 順次実行と変数 例題と演習問題の解答例

## 例題1

|  |
| --- |
| document.write("こんにちは<br>");  document.write("はじめての<br>");  document.write("プログラミング<br>"); |

### 【解説】

document.write ()関数を３回使っています。「こんにちは」「はじめての」「プログラミング」の順に表示するために、document.write("こんにちは<br>")、document.write("はじめての<br>")、document.write("プログラミング<br>")の順に書きます。

## 例題2

|  |
| --- |
| let a = "Hello";  a = "こんにちは";  alert(a); |

### 【解説】

「こんにちは」と表示されます。

変数aに最初は"Hello"を代入します。次に"こんにちは"と代入した後、変数aを表示するので、表示は「こんにちは」となります。変数には、最後に代入した値だけが残ることを確認しましょう。

※後の章で出てくる配列では、一つの配列に複数の値を代入することができます。

## 例題3

|  |
| --- |
| let a = 34567;  let b = 123456;  let c = a + b;  alert("a + b = " + c); |

### 【解説】

変数aに34567、変数bに123456を代入します。aとbの足し算の結果をcに代入します。最後に、計算式の文字列と、変数cを表示します。複数の文字列や変数の値をまとめて一つの文字列として扱いたい場合は、+(プラス)の演算で文字列連結を行えます。なお、右辺と左辺の値が両方とも数値であれば足し算が行われます。

## 例題4

|  |
| --- |
| let a = prompt("数字を入力してください");  alert(a); |

### 【解説】

関数prompt()を使って、ユーザーに入力をうながします。「数字を入力してください」と表示するには、引数として文字列を関数input()に渡します。

ユーザーが入力した値は、関数prompt()が受け取り、戻り値としてプログラムに返します。上の回答例では、戻り値を変数aに代入しています。

最後に関数alert()に変数aを渡して、表示します。

## 問題集1

|  |
| --- |
| alert("おはよう");  alert("こんにちは");  alert("こんばんは"); |

### 【解説】

alert()関数を３回使っています。「おはよう」「こんにちは」「こんばんは」の順に表示するために、alert ("おはよう")、alert ("こんにちは")、alert ("こんばんは")の順に書きます。

## 問題集2

|  |
| --- |
| let myouji = prompt("名字を入力してください");  let namae = prompt("名前を入力してください");  alert(myouji + namae); |

### 【解説】

変数myoujiと、変数namaeにそれぞれ値を代入します。上の例では変数myoujiに"山田"、変数namaeに"太郎"を代入しています。

関数alert()に、引数として変数myouji、namaeを渡します。複数の引数を渡すときには、『,』で区切ります。

## 問題集3

|  |
| --- |
| let a = parseInt( prompt("数字を入力してください") );  let b = a + 10;  alert("a + 10 = " + b); |

### 【解説】

関数prompt()にユーザーに数字を入力するよう促すメッセージを渡します。関数prompt()が返す値を、関数parseInt()に渡し、関数parseInt()の戻り値を変数aに代入します。

関数alert()の引数のカッコの中で、a+10の計算を行い、結果を表示させています。

第2章 条件による選択（分岐） 例題と演習問題の解答例

## 例題1

|  |
| --- |
| let a = parseInt( prompt("0か1を入力してください") );  if (a == 1){  document.write("1が入力されました");  } else {  document.write("1以外が入力されました");  } |

### 【解説】

関数prompt()を使い、ユーザーに0か1を入力するように促します。関数parseInt()を使い、入力された値を整数に変換して、変数aに代入します。

キーワードifで、変数aと1を比較します。演算子『==』を使うことで、左辺と右辺が等しいかどうか調べます。等しければ真（true）、等しくなければ偽（false）が返されます。

真（true）が返されたら「1が入力されました」と表示します。キーワードelseを使って、偽（false）が返されたときには「1以外が入力されました」と表示するようにします。

なお、値を代入するときの記号『=』は一つです。一方、比較するときは『==』と、二つになることに注意してください。

## 例題2

|  |
| --- |
| let saikoro = Math.floor( Math.random() \* 6 ) + 1;  document.write("サイコロ:" + saikoro + "<br>"); |

### 【解説】

関数Math.random()の値とサイコロの一番大きい値である6を掛け算して小数点を関数Maht.floor()で切り捨てることによりサイコロの値を生成します。生成された乱数は戻り値として返されるので、変数saikoroに代入します。

document.write ()で説明用の文字列とともに変数saikoroを表示します

## 問題集1

|  |
| --- |
| let a = parseInt( prompt("数字を入力してください") );  if ( a % 2 == 0 ){  alert("これは偶数です");  } else {  alert("これは奇数です");  } |

### 【解説】

関数prompt()を使い、数字を入力するように促します。関数parseInt()を使い、入力された値を整数に変換して、変数aに代入します。

キーワードifで、変数aを2で割ったときの余りと、0を比較します。演算子『%』を使うことで、2で割った余りを求めることができます。演算子『==』を使うことで、左辺と右辺が等しいかどうか調べます。等しければ真（true）、等しくなければ偽（false）が返されます。

## 問題集2

|  |
| --- |
| let a = Math.floor( Math.random() \* 6 ) + 1;  if (a == 1){  alert("一が出ました");  } else if (a == 2){  alert("二が出ました");  } else if (a == 3){  alert("三が出ました");  } else if (a == 4){  alert("四が出ました");  } else if (a == 5){  alert("五が出ました");  } else if (a == 6){  alert("六が出ました");  } |

### 【解説】

この問いはサイコロをプログラミングで再現する問題です。関数Math.random()の値とサイコロの一番大きい値である6を掛け算して小数点を関数Maht.floor()で切り捨てることによりサイコロの値を生成します。生成された乱数は戻り値として返されるので、変数aに代入します。

キーワードifを使い、変数aと1が等しいか比較します。等しく無い場合、次のキーワードelse ifの比較が実行されます。2と等しいか、3と等しいか、…と、6までと等しいか比べます。

等しい値が見つかったら、関数alert()を使って漢数字を表示します。

第3章 配列 例題と演習問題の解答例

## 例題1

|  |
| --- |
| let a = ["A", "B", "C", "D"];  alert( a[3] ); |

### 【解説】

"D"が表示されます。

リストに四つの要素を設定し、添え字（インデックス）を指定して、要素を表示します。添え字は0から始まることに注意しましょう。0番目が"A"、1番目が"B"、2番目が"C"、3番目が"D"です。

## 例題2

|  |
| --- |
| let a = ["A", "B", "C", "D"];  alert(a); |

【解説】

関数alert()にリストaを渡すと、全ての要素を表示します。

リストの要素を指定するための記号『 [ ] 』や、添え字を指定する必要はありません。

## 例題3

|  |
| --- |
| let a = ["A", "B", "C"];  a[0] = "a";  alert(a); |

### 【解説】

"a"、"B"、"C"と表示されます。先頭の要素（0番目の要素）が小文字の"a"になっていることを確認しましょう。

プログラムは、リストaに３つの要素を設定した後、添え字に0を指定して、値を代入しています。

## 例題4

|  |
| --- |
| let aisatsuList = [];  aisatsuList.push("おはよう");  aisatsuList.push("こんにちは");  aisatsuList.push("こんばんは");  alert(aisatsuList); |

### 【解説】

空の配列aisatsuListを用意しています。aisatsuList.push ()を3回呼び出して、「おはよう」「こんにちは」「こんばんは」を配列に追加しています。最後のalert(aisatsuList) で、aisatsuList.push ()が呼び出された順に要素が追加されていることが確認できます。

## 例題5

|  |
| --- |
| let aisatsuList = [];  aisatsuList.push("おはよう");  aisatsuList.push("こんにちは");  aisatsuList.push("こんばんは");  aisatsuList.splice(1, 1);  alert(aisatsuList); |

### 【解説】

「おはよう」「こんばんは」と表示されます。

プログラムの冒頭では、空の配列aisatsuListを用意しています。aisatsuList.push ()を3回呼び出して、「おはよう」「こんにちは」「こんばんは」を配列に追加しています。続いて、aisatsuList.splice (1, 1)を実行し、1番目の要素を取り除きます。

最後のalert( aisatsuList ) で、aisatsuList.push()が呼び出された順に要素が追加されたこと、1番目の要素「こんにちは」が削除されていることが確認できます。

## 問題集1

|  |
| --- |
| let a = ["A", "B"];  document.write("a[1]:" + a[1]);  document.write("<br>");  a.push("C");  document.write("a[1]:" + a[1] + "<br>");  document.write("a[2]:" + a[2] + "<br>");  document.write("<br>");  a.splice(2,0,"b");  document.write("a[1]:" + a[1] + "<br>");  document.write("a[2]:" + a[2] + "<br>");  document.write("<br>");  b = a.pop();  document.write("a[1]:" + a[1] + "<br>");  document.write("a[2]:" + a[2] + "<br>");  document.write("b:" + b); |

### 【解説】

最初はa[1]: Bと表示されます。

続いて、a[1]: B、a[2]: C と表示されます。添え字は0から始まることと、"C"が追加されたことを確認しましょう。

次のa.splice(2, 0, "b")は、"C"があった場所に"b"を挿入します。"C"は後ろにずれます。a[1]: B、a[2]: bと表示されます。

次のa.pop()で、"C"は削除されます。表示はa[1]: B、a[2]: bで変更ありません。

a.pop()は削除した値を戻り値として返すので、変数bには"C”が代入されます。これにより、最後のprint("b:", b)で "C"が表示されるわけです。

## 問題集2

|  |
| --- |
| let b = [];  b.push(1);  b.push(2);  b.push(3);  b.push(4);  b.push(5);  document.write( b.splice(0, 1) + ",");  document.write( b.splice(0, 1) + ",");  document.write( b.splice(0, 1) + ",");  document.write( b.splice(0, 1) + ",");  document.write( b.splice(0, 1) );  alert(b); |

### 【解説】

空の配列bを用意し、b.push()を5回実行して、1から5まで要素を追加しています。続いて、b.splice(0, 1)で要素を取り出し、取り出した要素をdocument.write()で表示しています。b.splice(0, 1)は、0番目の要素以降から1つ削除しながら取り出すため、リストの要素は１つずつ前に詰められます。この命令を5回実行すると、配列から全ての要素が削除されることになります。空の配列bをalert()に渡すと、空をあらわす表示になります。

第4章 繰り返し 例題と問題集の解答例

## 例題1

|  |
| --- |
| let h = [1, 2];  for (let i = 0; i < h.length; i++){  document.write(h[i] + "<br>");  }  document.write("おわり<br>"); |

### 【解説】

1、2と表示した後、最後に「おわり」と表示します。元のプログラムでは繰り返しごとに「おわり」と表示していましたが、繰り返しが終わった後に表示することを確認してください。

## 例題2

|  |
| --- |
| for (let i = 1; i < 5; i++){  document.write(i + "<br>");  } |

【解説】

for文にて繰り返し処理を行います。まず変数iを1で初期化、iが5未満のあいだ繰り返し、繰り返すたびにiの値を1増やします。繰り返し処理ではdocument.write()を用いて変数iを表示します。変数iの表示を繰り返すことで、1、2、3、4と順に表示できます。

## 例題3

|  |
| --- |
| for (let i = 4; i > 0; i = i - 1){  document.write(i + "<br>");  } |

### 【解説】

for文にて繰り返し処理を行いますまず変数iを4で初期化、iが0より大きいあいだ繰り返し、繰り返すたびにiの値を1減らします。繰り返し処理ではdocument.write()を用いて変数iを表示します。変数iの表示を繰り返すことで、4,3,2,1と順に表示できます。

## 例題4

|  |
| --- |
| let a = Math.floor( Math.random() \* 10 ) + 1;  let i = 0;  while ( i < a ){  document.write("a:" + a + "<br>");  i = i + 1;  } |

### 【解説】

関数Math.random()とMath.floor()を使い、1～10の範囲の整数を作り変数aに代入します。次に変数iを0で初期化します。そしてwhile文を用いて変数iが変数aより小さいあいだ繰り返し処理を行います。繰り返し処理では変数aの値を表示しつつ、変数iの値を一つ増やします。変数aの表示は、変数aの値と同じ回数、繰り返されます。

## 例題5

|  |
| --- |
| let a = Math.floor( Math.random() \* 10 ) + 1;  for (let i = 0; i < a; i++) {  document.write("a:" + a + "<br>");  } |

### 【解説】

例題4をfor文で再現した内容になります。for文で変数iの初期化、変数iとaの比較、iの値を一つ増やす処理を行っています。変数iを1つ増やす処理はi = i + 1と記述する他、i++やi += 1と記述することもできます。

## 問題集1

|  |
| --- |
| for (let i = 11; i <= 20; i++){  document.write(i + "<br>");  } |

### 【解説】

for文にて繰り返し処理を行います。変数iを11で初期化、iが20以下のあいだ繰り返し、繰り返すたびに変数iの値を一ずつ増やします。繰り返し処理では変数iの値を表示します。

## 問題集2

|  |
| --- |
| for (let i = 0; i <= 100; i++){  if ( i % 3 == 0 ){  document.write(i + "<br>");  }  } |

### 【解説】

for文にて繰り返し処理を行います。変数iを0で初期化、iが100以下のあいだ繰り返し、繰り返すたびに変数iの値を一ずつ増やします。繰り返し処理では更にif文にて条件分岐を行います。変数iを3で除算し余りが0の時だけ変数iの値を表示します。

## 問題集3

|  |
| --- |
| let a = Math.floor( Math.random() \* 6 ) + 1;  b = 0;  while ( a > 0 ) {  b = b + a;  a = a - 1;  }  document.write("b:" + b); |

### 【解説】

関数Math.random()とMath.floor()を使い、1～6の範囲の整数を作り変数aに代入します。

変数bに0を代入します。

while文で変数aが0より大きい間、繰り返すような条件の式を書いてます。

whileの繰り返しの中で、b=b+a、a=a-1の計算を行います。

whileの繰り返しの後に変数bの値を表示しています。

第5章 関数の定義と利用 例題と問題集の解答例

## 例題1

|  |
| --- |
| aisatsu = prompt("なにか言葉を入力してください:");  document.write(aisatsu); |

### 【解説】

関数prompt()を使い、「なにか言葉を入力してください:」と表示し、入力を求めます。ユーザーが入力した値は、関数prompt()の戻り値として返され、変数aisatsuに代入されます。その変数aisatsuを関数document.write()で表示します。

これでユーザーからの入力を受け取り、受け取ったまま表示することができました。

## 例題2

|  |
| --- |
| let hiduke\_jikoku = new Date();  nen = hiduke\_jikoku.getFullYear();  tsuki = hiduke\_jikoku.getMonth() + 1;  hi = hiduke\_jikoku.getDate();  alert(nen + "年" + tsuki + "月" + hi + "日"); |

### 【解説】

最初に、現在の日付・時刻をhiduke\_jikoku変数に代入します。次に年・月・日を取り出しそれぞれ変数に代入。最後に関数alert()で表示しています。

## 例題3

|  |
| --- |
| function aisatsu() {  alert("こんにちは");  }  aisatsu();  aisatsu(); |

### 【解説】

キーワードfunctionを使って、関数aisatsu()を定義します。関数の定義の中での処理を決めるためalert ()を書きます。

定義した関数aisatsu()を2回呼び出すと、2回「こんにちは」と表示されます。1回も呼び出さなければ、「こんにちは」と表示はされません。関数の定義の中にalert()と書いてあっても、関数が呼び出されなければ、alert()は実行されません。

## 例題4

|  |
| --- |
| function shouhizei(kingaku) {  zeigaku = kingaku \* 0.1;  return zeigaku;  }  alert("消費税額" + shouhizei(12000), "円"); |

## 【解説】

キーワードfunctionを使って、関数shouhizei()を定義します。関数を呼び出すときに、引数として金額を渡すため、後ろの小かっこの中にkingakuと書いて、引数を定義します。

関数の中で、そのkingakuを使った計算を行い、returnで計算結果を呼び出し元に返すようにしています。

実際に呼び出すときは、関数名(引数の値）とします。上の例では、「shouhizei( 12000 )」が関数を呼び出している部分です。

## 問題集1

|  |
| --- |
| aisatsu = ["おはよう", "こんにちは", "こんばんは"];  rand = Math.floor (Math.random() \* 3);  alert(aisatsu[rand]); |

### 【解説】

配列aisatsuに、三つのあいさつの言葉を代入しています。

変数randに0～2のいずれかの値をランダムに代入し、配列aisatsuの添え字に使って、値を表示します。

## 問題集2

|  |
| --- |
| function shouhizei(kingaku, zeiritsu) {  zeigaku = kingaku \* zeiritsu / 100;  return zeigaku;  }  alert("消費税額" + shouhizei(12000, 10), "円"); |

### 【解説】

キーワードfunctionを使って、関数zeigaku\_keisan()を定義します。この関数では、金額と税率の二つの引数を取ることにするので、小かっこの中に二つ（kingaku, zeiritsu）引数を書いています。カンマ（ , ）で区切っていることも確認してください。

計算した結果をキーワードreturnで返すように定義しています。

呼び出す際にはzeigaku\_keisan(12000, 10)として、計算させたい値を二つ渡しています。値を渡すときにも、カンマで区切っています。

関数から返された値を変数kekkaに代入し、関数alert() で表示させています。

第6章 繰り返し（反復）と選択（分岐）の組み合わせ　 例題と問題集の解答例

## 例題1

|  |
| --- |
| for (let i = 1; i < 31; i++) {  if (i % 3 == 0) {  document.write(i,"は3の倍数です<br>");  }  } |

### 【解説】

for文にて繰り返し処理を行います。変数iを0で初期化、iが31未満のあいだ繰り返し、繰り返すたびに変数iの値を一ずつ増やします。繰り返し処理では更にif文にて条件分岐を行います。変数iを3で除算し余りが0の時だけ変数iの値を表示します。

## 例題2

|  |
| --- |
| aisatsu = ["おはよう", "こんにちは", "こんばんは"];  for (let i = 1; i < 6; i++) {  for (let j = 0; j < aisatsu.length; j++) {  document.write( aisatsu[j] ,"<br>");  }  } |

### 【解説】

あいさつ配列を作ります。次に、for文にて繰り返し処理を行います。変数iを1で初期化、iが6未満のあいだ繰り返し、繰り返すたびに変数iを一ずつ増やします。繰り返しの中であいさつ配列の値をもとに更に繰り返し処理を行います。あいさつ配列の要素全てについて、表示処理を繰り返すことになるので、3つのあいさつの言葉を5回繰り返すプログラムになります。

## 例題3

|  |
| --- |
| let kotae = 0;  for (let n = 1; n < 10; n++){  document.write(n + "の段:");  for (let i = 1; i < 10; i++) {  kotae = n \* i;  if (kotae > 25){  break;  }  }  document.write(kotae + " ");  document.write("<br>");  } |

### 【解説】

もとのプログラムは二重ループの2段目でdocument.write()を使い各段の値を列挙していた。今回のプログラムは二重ループの1段目かつ2段目のfor文の次でdocument.write()を使っているため、各段の最後のだけが表示される。

なお、if文により変数kotaeが25を越えた時点でbreakするため、列挙される値は常に25より大きい値が表示される。

## 問題1

|  |
| --- |
| let aisatsu = ["おはよう", "こんにちは", "こんばんは"];  for (let i = 0; i < aisatsu.length; i++) {  if (i % 2 == 0) {  document.write(aisatsu[i]);  }  } |

### 【解説】

あいさつ配列を作ります。次に、for文で配列の要素の数だけ繰り返し処理を行います。if文で変数iが2で割り切れる（偶数）の時だけ、document.wirte()を実行します。これにより添え字が偶数のあいさつ配列の要素だけが表示されます。

## 問題2

|  |
| --- |
| for (let i = 9; i > 0; i--) {  document.write(i + "の段<br>");  for (let j = 9; j > 0; j--){  document.write( i \* j + "<br>");  }  } |

### 【解説】

for文にて繰り返し処理を行います。変数iを9で初期化、iが0より大きいあいだ繰り返し、繰り返すたびに変数iの値を一ずつ減らします。繰り返し処理では更にfor文にて繰り返し処理を行います。for文にて繰り返し処理を行います。変数jを9で初期化、jが0より大きいあいだ繰り返し、繰り返すたびに変数jのあたいを一ずつ減らします。繰り返し処理では変数iとjを掛け算した結果を表示することで、九九を後ろから表示するプログラムになります。

## 問題3

|  |
| --- |
| function dan\_keisan(n){  for (let j = 9; j > 0; j--){  document.write( n \* j + "<br>");  }  }  for (let i = 9; i > 0; i--) {  document.write(i + "の段<br>");  dan\_keisan(i);  } |

### 【解説】

関数dan\_keisan(n)を定義します。関数を呼び出すときに段数を渡すため、引数nを定義します。for文にて繰り返し処理を行います。変数jを9で初期化、jが0より大きいあいだ繰り返し、繰り返すたびに変数jのあたいを一ずつ減らします。繰り返し処理では変数nとjを掛け算した結果を表示します。

関数宣言に続き、for文にて繰り返し処理を行います。変数iを9で初期化、iが0より大きいあいだ繰り返し、繰り返すたびに変数iの値を一ずつ減らします。繰り返し処理では関数dan\_keisan()を呼び出し引数に変数iを渡します。

dan\_keisanをfor文で繰り返し呼ぶことで結果として問題2と同じ出力になります。