



JavaScriptを使った正弦波の作図プログラム

名前: _____

1. やってみよう HTML で文字を表示してみよう。

HTMLとは、Hyper Text Markup Language（ハイパーテキスト・マークアップ・ランゲージ）の略で、Webページを作るために開発された、最も基本的なマークアップ言語です。

ファイルLesson1-1をダブルクリックしてみましょう。ダブルクリックするとWebブラウザが開き、画面に「Hello」と文字が表示されます。では、このファイルを、テキストエディタで開いてみましょう。画面に文字を表示するHTMLファイルの中身を見ることができます。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8">
  <title>文字の表示</title>
</head>
<body>
  Hello
</body>
</html>
```

Lesson1-1.文字の表示

演習1 テキストエディタで Lesson1-1 に示した「Hello」の部分をお自分の名前に変更して上書き保存し、Webブラウザで確認してみよう。

2. やってみよう JavaScript で計算をしてみよう。

JavaScriptはスクリプト言語と呼ばれる種類の、手軽に記述できるプログラミング言語です。HTMLファイル内のscriptエレメント（<script>~</script>）部分に記述することで、HTMLファイルがWebブラウザに読み込まれると、WebブラウザがJavaScriptのプログラムを解釈して実行してくれます。

演習2 Lesson1-1 のプログラムに、Lesson1-2 に示した script エレメントを追加して、簡単な計算式を作ってみよう。

演習3 半径が5cmの円の面積を求めるプログラムに変更してみよう。

JavaScriptでは「 π 」を「Math.PI」と入力し計算します。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8">
  <title>文字の表示</title>
</head>
<body>
  みやぎ たろう<br>
  <script>
    var a = 10;
    var b = 20;
    var c = a + b;
    document.write( a, "+", b, "=", c);
  </script>
</body>
</html>
```

Lesson1-2.JavaScriptで計算

3. やってみよう WebAPI を使った正弦波のグラフを見てみよう。

APIとはApplication Programming Interface（アプリケーション・プログラミング・インターフェイス）の略で、ある1つの機能に特化した公開されているプログラムです。作りたい機能がAPIで公開されていれば、同じプログラムを一から作る必要がなく、時間とコストを大幅に削減できます。また、Web上に公開されたAPIをWebAPIと呼び、外部から呼び出して利用できます。WebAPIは、さまざまな分野で公開されていますが、その多くは無料で利用が可能です。

ここでは、canvas要素を指定してグラフを表示するChart.jsというWebAPIを利用してみます。デフォルトでアニメーションなどが使え、特に設定を作りこまなくても比較的整ったグラフが作成できます。Lesson2-1をWebブラウザとテキストエディタで開いてWebAPIを確かめてみよう。

なお、今回使用するWebAPIは、Internet Explorerでは動作しない場合があります。プログラムを実行するブラウザにはGoogleChromeまたはEdgeを推奨します。

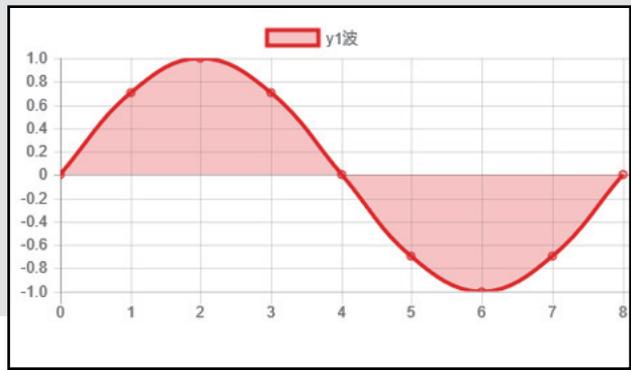
```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title>波の合成</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.4/Chart.min.js"></script>
<canvas id="myChart" width="1000" height="500"></canvas>
</head>
<body>
<script>
var ctx = document.getElementById("myChart").getContext("2d");
var myChart = new Chart(ctx, {
  type: "line",
  data: {
    labels: [0, 1, 2, 3, 4, 5, 6, 7, 8],
    datasets: [{
      label: "y1波",
      data: [0, 0.7, 1, 0.7, 0, -0.7, -1, -0.7, 0],
      backgroundColor: "rgba(230, 10, 10, 0.2)",
      borderColor: "rgba(230, 10, 10, 1)"
    }]
  }
});
</script>
</body>
</html>

```

グラフを表示するWebAPIのプログラムを取り込む設定をする

グラフ(チャート)の設定には様々なオプションがあります。下のサイトを参考にしてください。



Lesson2-1.WebAPIを使った正弦波の表示プログラム

演習4 「data: [0, 0.7, 1, 0.7, 0, -0.7, -1, -0.7, 0]」の数値を変えて、Webブラウザで確認してみよう。

演習5 backgroundColor や borderColor の数値を変えて、Webブラウザで確認してみよう。

() 内の数値は0~255までの間で色の濃さを調整できます。それぞれ(赤、緑、青、透過度)の順で濃さを変えることにより、様々な色を表現できます。

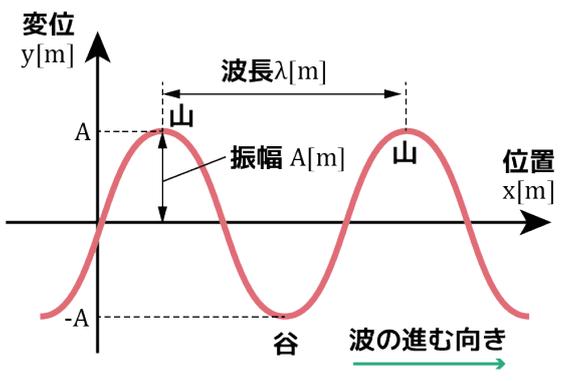
4.考えてみよう 正弦波を表す式を考えてみよう。

x軸を正の向きに伝わる正弦波について、振幅をA[m]、時刻をt[s]、周期をT[s]、位置をx[m]、波長をλ[m]としたとき、正弦波の変位y[m]を表す式を考えてみよう。

y =

上の式をプログラムの式に置き換えてみよう。
 sinX → Math.sin(X) λ→ram に置き換える

y =



5. やってみよう 正弦波の式を使ってグラフを作ってみよう。

演習6 Lesson2-1のグラフを「正弦波の式」で置き換えてみよう。

正弦波の各値は、振幅 $A=1[m]$ 、周期 $T=10[s]$ 、波長 $\lambda=24[m]$ 、時刻 $t=0[s]$ とします。また、媒質の位置 $x[m]$ を0から24まで変化させて変位 $y1[m]$ を計算します。

script部分のみ

```
<script>
var ctx = document.getElementById("myChart").getContext("2d");
var myChart;
var label = new Array();
var y1Array = new Array();
var A = 1;
var T = 10;
var x = 0;
var ram = 24;
var t = 0;
for (x = 0; x <= 24; x++) {
  label[x] = x;
  var y1 = A * Math.sin(2 * Math.PI * (t / T - x / ram));
  y1Array[x] = y1;
}
myChart = new Chart(ctx, {
  type: "line",
  data: {
    labels: label.slice(0),
    datasets: [{
      label: "y1波",
      data: y1Array.slice(0),
      backgroundColor: "rgba(230, 10, 10, 0.2)",
      borderColor: "rgba(230, 10, 10, 1)"
    }]
  }
});
</script>
```

グラフに使用する変数や配列の準備する

繰り返し文を使い、変位 y の計算結果を、配列に入れていく

x軸に配列「label」の部分配列を取得する

y軸に配列「yArray」の部分配列を取得する

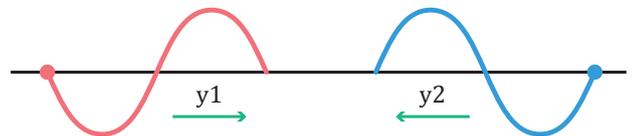
Lesson2-2.APIを使った正弦波の表示プログラム例

6. やってみよう 2つの正弦波のグラフを作ってみよう。

演習7 x軸を負の向きに進む正弦波 $y2$ の式を考え、プログラムに追加してみよう。

計算結果の配列は $y2Array$ とし、振幅 A 、周期 T 、波長 λ は、 $y1$ と同じ変数の値を使用します。また、媒質の位置 $x[m]$ は2周期分を表示させたいのでfor文の条件式の部分を「24」から「48」に変更しましょう。

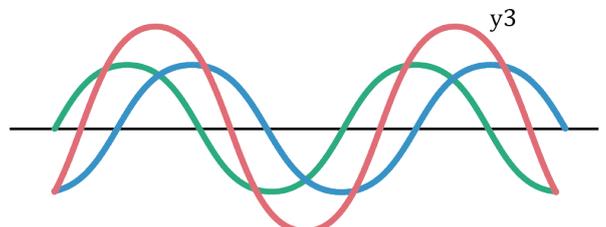
$y2 =$



7. やってみよう 合成波のグラフを作ってみよう。

演習8 合成波 $y3$ を求める式を考え、プログラムに追加してみましょう。計算結果の配列は $y3Array$ とします。

$y3 =$



8. やってみよう 2つの正弦波を動かしてみよう。

演習9 Lesson2-2に、位相を変えるボタンを追加してy1とy2が逆向きに進むように変更してみよう。

body + script部分のみ

```
<body>
<input type="button" value="進む" onclick="draw()">
<script>
var ctx = document.getElementById('myChart').getContext('2d');
var myChart;
var y1Array = new Array();
var y2Array = new Array();
var y3Array = new Array();
var label = new Array();
var A = 1;
var T = 10;
var x = 0;
var ram = 24;
var t = 0;
function draw(){
  if(myChart){
    myChart.destroy();
  }
  for (x = 0; x <= 48; x++) {
    label[x] = x;
    var y1 = A * Math.sin( 2 * Math.PI * ( t / T - x / ram));
    var y2 = A * Math.sin( 2 * Math.PI * ( t / T + x / ram));
    var y3 = y1 + y2;
    y1Array[x] = y1;
    y2Array[x] = y2;
    y3Array[x] = y3;
  }
  t += 0.1;
  myChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: label.slice(0),
      datasets: [{
        label: 'y1波',
        data: y1Array.slice(0),
        backgroundColor: "rgba(0, 0, 0, 0)",
        borderColor: "rgba(230, 10, 10, 1)"
      },
      {
        label: 'y2波',
        data: y2Array.slice(0),
        backgroundColor: "rgba(0, 0, 0, 0)",
        borderColor: "rgba(10, 230, 10, 1)"
      },
      {
        label: '合成波(y1+y2)',
        data: y3Array.slice(0),
        backgroundColor: "rgba(0, 0, 0, 0.2)",
        borderColor: "rgba(10, 10, 230, 1)"
      }
    ]
  },
  options: {
    animation: false ,
    responsive: false ,
    scales:
      {yAxes:[
        {ticks: {
          min: -3,max: 3
        }}
      ]}
  }
});
draw();
</script>
</body>
```

「進む」ボタンでグラフを動かすためにグラフデータを関数化する

一つ前のグラフデータを削除する

正弦波の位相をずらすため変数 t を0.1増やす

グラフのオプション設定
アニメーションの停止
y軸の最大値、最小値の設定

グラフ表示関数の実行

9.考えてみよう グラフから正弦波の性質を考えてみよう。

y_1 と y_2 の位相が「重なったとき」と「反対のとき」の合成波 y_3 はどのようなになっているか説明してみよう。

y_1 と y_2 が反対方向に動いているとき、合成波 y_3 はどのような動きをしているか説明してみよう。

波の波長 λ [m]が異なる場合、合成波はどのようなになるか説明してみよう。

10.考えてみよう 音波を使って合成波の性質を体験してみよう。

波の性質について理解を深めるために、次のプログラムを体験してみよう。

① 音波と正弦波	② トーン信号と合成波	③ 母音と合成波
		