明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

CMP実習2
JavaScriptの基本

中村、宮下、斉藤、菊池

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

Webでユーザの操作を取得

- テキスト入力していないのに送信ボタンを押さないで！
- ユーザが入力する前には入力例を示しておいて，入力開始しようとしたら消したい！
- リアルタイムに値を取得して表示したい！
- ユーザが地図上で操作をしたら，表示内容を変更したい！

JavaScript

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

JavaScript超入門

- イベントが発生した時にどうするのかを記述する言語
  - HTMLの様々な部品に対してイベントを追加していく
    - 画像にマウスがホバーされたら
    - ボタンがクリックされたら
    - 入力フォームからマウスが外れたら

    などなど
- HTML内部で何らかのユーザ操作が行われた時に，リロードせずにページ上で動作するためのもの
  - クリックされた！じゃあ…
  - 文字入力された！じゃあ…
  - マウスカーソルが上に来た！じゃあ…
  - マウスカーソルがどこかに行った！じゃあ…

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

つまり…

- 必要な知識
  - HTMLのどの部品なのかを指定する方法
  - その部品にイベントを埋め込む方法
  - それをJavaScriptで記述する方法

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

ProcessingとJavaScript

- 同じ所: プログラミング言語なので変数，計算，条件分岐，繰返し，メソッド，クラスなど同じ

12345の約数の数を数えるプログラム

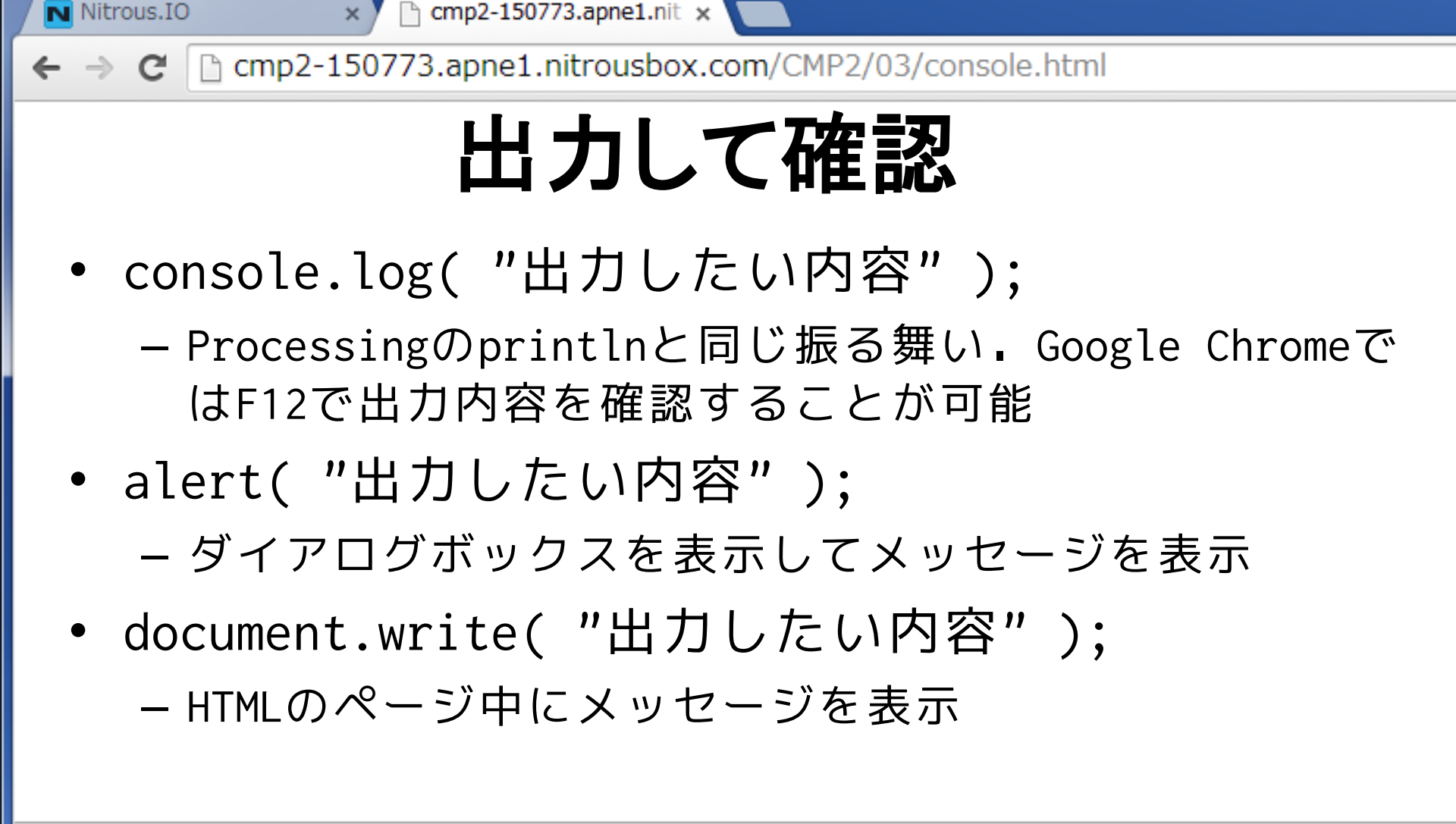
```
int i = 1;
int count = 0;
while( i <= 12345 ){
  if( (12345 % i) == 0 ){
    // 12345をiで割った余りが
    // 0だったらcountを増やす
    count++;
  }
  i++;
}
println( "約数の数は"+count );
```

```
<script>
var i = 1;
var count = 0;
while( i <= 12345 ){
  if( (12345 % i) == 0 ){
    // 12345をiで割った余りが
    // 0だったらcountを増やす
    count++;
  }
  i++;
}
alert( "約数の数は"+count );
</script>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

JavaScript超入門

- スクリプトタグで囲う
- 変数の定義は var 変数名
  - intとかfloatとかの型はない（何でも一緒）
- 行の最後にはセミコロン
- 関数の定義は function 関数名(引数);
  - 引数は引数名だけで良い（varは不要）
  - 返り値は return で！（一緒）
- 変数とか文字列をくっつける際は「+」
- ifとかforとかwhileとかは一緒


Nitrous.IO
cmp2-150773.apne1.nit
cmp2-150773.apne1.nitrousbox.com/CMP2/03/console.html
出力して確認
• console.log("出力したい内容");
– Processingのprintlnと同じ振る舞い. Google Chromeではf12で出力内容を確認することが可能
• alert("出力したい内容");
– ダイアログボックスを表示してメッセージを表示
• document.write("出力したい内容");
– HTMLのページ中にメッセージを表示
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
test
Hello world
中村聡史
50

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

もし～だったら

```
if( 条件A ){
    条件Aのときの動作
} else if( 条件B ){
    条件Bのときの動作
} else {
    どの条件にもあわなかったときの動作
}
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演算子	意味
x > y	x が y より大きい
x < y	x が y より小さい
x >= y	x が y 以上
x <= y	x が y 以下
x == y	x と y が等しい
x === y	x と y が等しい（厳密）
x != y	x と y が等しくない
x !== y	x と y が等しくない（厳密）
!x	x は false

演算子	意味
&&	かつ
||	または
!	true/false反転

```
console.log( 10 == '10' );
console.log( 10 === '10' );
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

配列とオブジェクト

- [] で囲まれた部分が配列の定義となる

```
var a = [ 1, 2, 3, 4, 5 ];
a[0] = 5;
console.log( a[1] );
```

- {} で囲まれた部分がオブジェクトの定義となる

```
var human = { name: "宮下", age: 40,
              position: "准教授" };
human.position = "教授";
console.log( human.name + human.position );
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

オブジェクト

- {} で囲まれた部分がオブジェクトの定義となる
- オブジェクトの各要素はコロン「:」で繋いで定義

要素名: 要素の値

オブジェクト名.要素名

```
var human =
{ name: "宮下", age: 40, position: "教授" };
```

```
var human = new Object();
human.name = "宮下";
human.age = 40;
human.position = "教授";
```

同じ意味

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

関数の定義

- 主に下記の2つのタイプが良く利用される（基本的に両者は同じ意味）

```
function hoge(){ ... }   （関数宣言）
hoge = function(){ ... };   （関数式）
```

- どちらの場合も下記のように呼び出すが，関数式の場合は，定義以降でしか呼び出せない

```
hoge();
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

JavaScriptの特徴

- オブジェクトに関数を追加可能

```
var human = {
    name: "宮下",
    position: "教授",
    drink: function(){
        alert( "エナジー！" );
    }
};
```

```
var human = {
    name: "宮下",
    position: "教授"
};
```

```
human.drink = function(){
    alert( "エナジー！" );
}c
```

human.drink();で実行可能

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

ユーザの行動に連動させる

- 要素にイベントと対応する関数を埋め込む！
  - マウスのボタンが押された時，カーソルが上に移動してきた時，テキストボックスにキー入力された時に何か実行させたい！

[方法]

[1] HTMLタグ（要素）にイベントと関数を埋め込む

```
<input type=”button” value=”OK” onclick="hoge()">
```

[2] 要素を取得してイベントと関数を割り当てる

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

要素へのイベントと関数の割り当て

ボタン . クリックされたら ➔ メッセージを表示

```
ボタン.クリックされたら = function(){
    メッセージを表示;
}
button.onclick = function(){
    alert( ”クリックされたよ！” );
}
```

```
human.drink = function(){
   alert( ”エナジー！” );
}
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

要素に付与可能なイベント一覧

- onload: 読み込みが終わった時
- onresize: ウインドウの幅が変わった時
- onclick: 要素上でクリックした時
- ondblclick: 要素上でダブルクリックした時
- onmouseover: 要素にマウスカーソルが乗った時
- onmouseout: 要素にマウスカーソルが乗った時
- onmousedown: 要素上でマウスのボタンを押した時
- onmouseup: 要素上でマウスのボタンを離した時
- onmousemove: 要素にマウスカーソルがのった時
- onkeydown: 要素上でキーボードのキーを押した時
- onkeyup: 要素上でキーボードのキーを離した時
- onfocus: 要素にフォーカスがあたった時
- onblur: 要素からフォーカスが外れた時
- onchange: フォームの値が変わった時
- onsubmit: フォームを送信する時

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

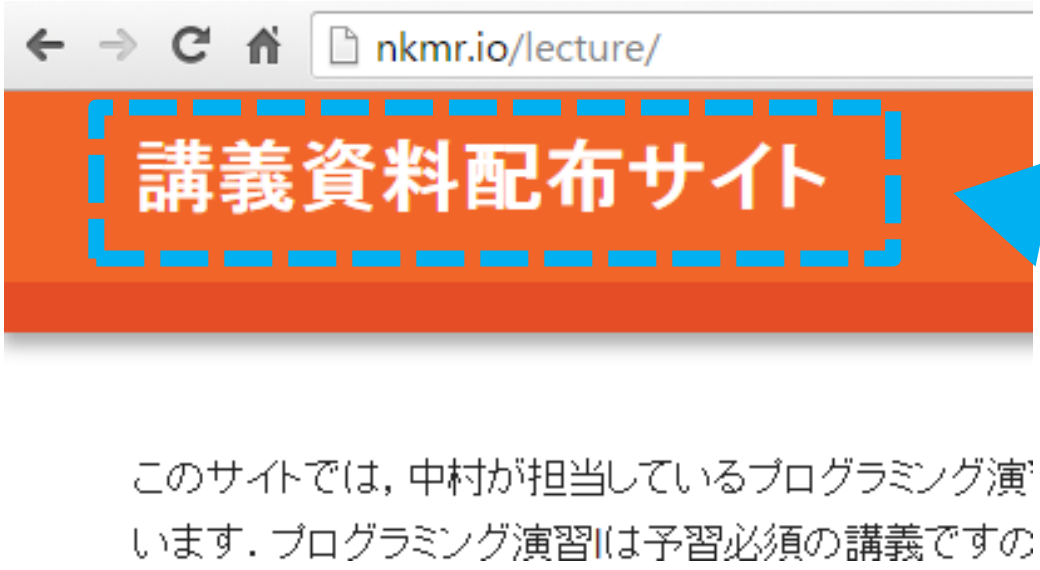
要素をどうやって取得する？

- DOMとはDocument Object Model
  - HTMLやXMLの各要素についてアプリケーションから利用するためのAPIのこと
  - HTMLやXMLの任意のタグの情報を取得したり，差し替えたりすることができる
- DOMツリーとは，HTMLやXMLの木構造情報
  - 木構造の情報

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演習

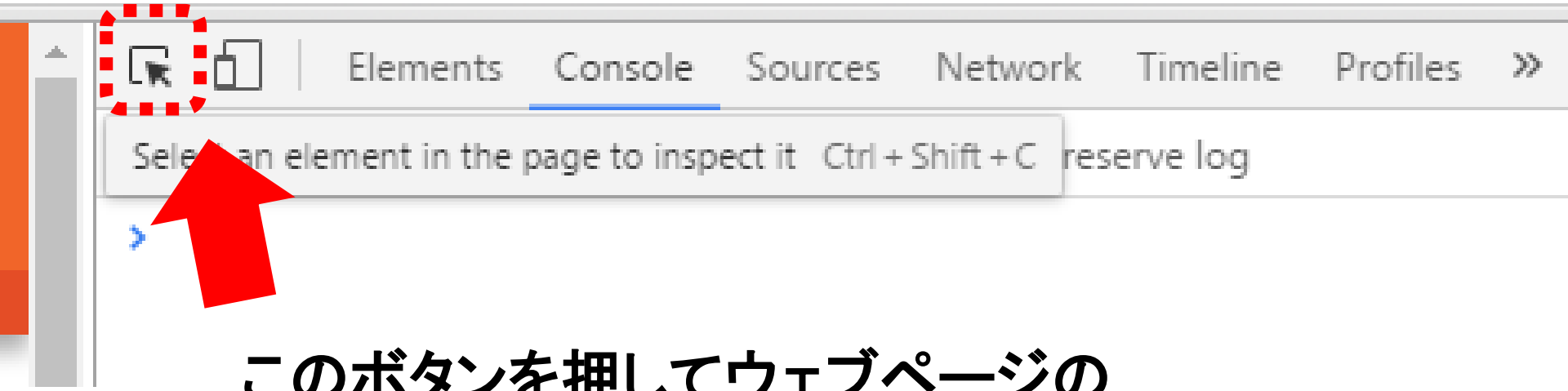
- http://nkmr.io/lecture/ にアクセスしましょう
- コンソールを開こう！



ここをクリックしたら
メッセージが表示される
ようにしたい！

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

要素をどうやって取得する？



このボタンを押してウェブページの
該当する部分をクリックしてみよう！

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

要素を取得してみよう

- クリックして要素を取得

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

要素をどうやって取得する？

- 1つの要素を取得
  - document.getElementById("id名");
  - document.querySelector("セレクタ名");
- 複数の要素を配列として取得
  - document.getElementsByClassName("class名");
  - document.getElementsByTagName("tag名");
  - document.querySelectorAll("セレクタ名");
- セレクタ名はCSSの表記方法
  - idは「#id名」，クラスは「.class名」，tagは「tag名」
  - 子の指定は「>」．「#id名 > tag名」のように指定

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

どうやって部品を取得する？

セレクタ名はこんな感じ

```
html > body > div.header-container >
                    header.wrapper.clearfix > h1.title
```

こうやって取得する！

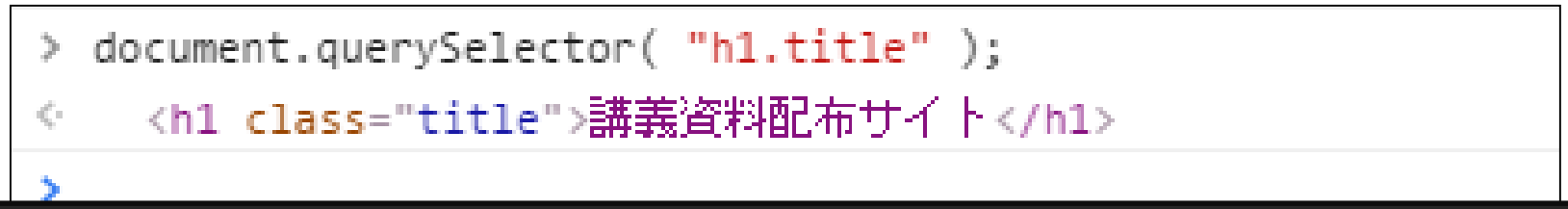
```
document.querySelector( “html > body > div.header-
container > header.wrapper.clearfix > h1.title” );
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

取得する方法

- コンソールにこんな感じで入力＋エンター

- 一意に識別できればいいので、下記でOK！




```
document.querySelector( “h1.title” );
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

コンソールの使い方

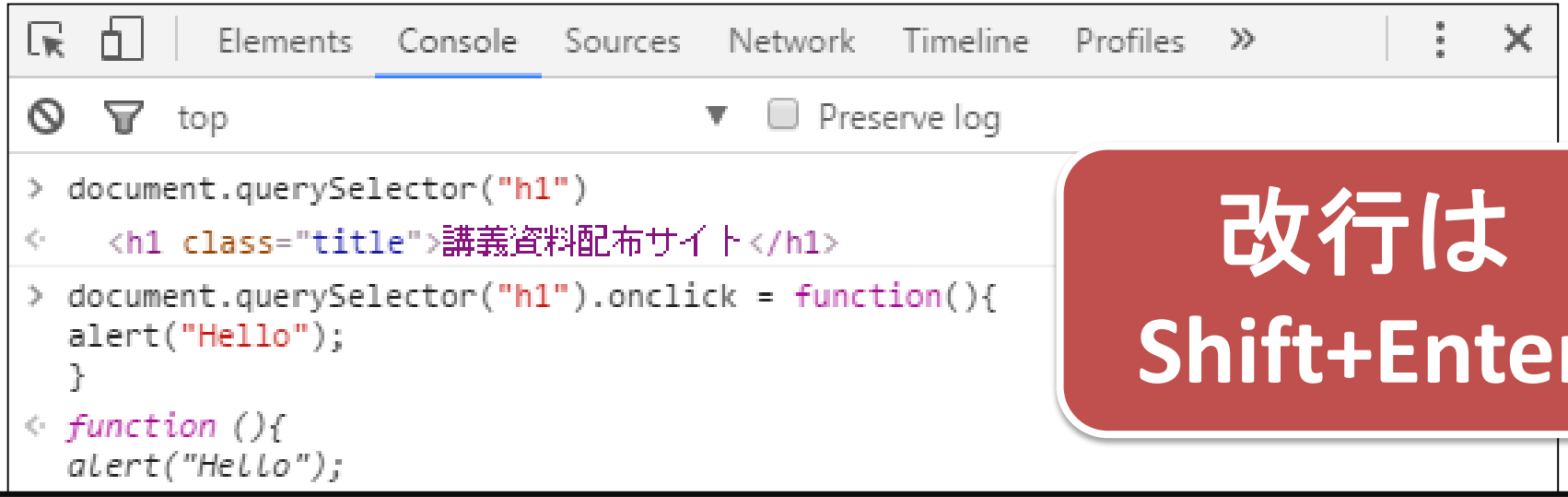
- 入力ミスしてしまったら，↑ボタンを押そう！
  - ↑ボタンを押すことで前に入力したものが再度表示され編集可能となります

- 候補が出てきたときにタブを押すと，それにより設定することが可能となります

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

とりあえず

- プログラムを埋め込んでみよう！
  - alertでメッセージを表示



改行は
Shift+Enter

```
document.querySelector( “h1” ).onclick = function(){
  alert( “Hello” );
}
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

DOM自体の操作

- 取得される要素はオブジェクト型
  - 要素 . 属性名 という形で，様々な情報へアクセス可能＋情報の差し替え可能

（例）var node = X.getElementById(何か);で取得

- テキスト入力ボックスの場合
  - `console.log( node.value ); // テキストの入力内容を取得`
- 一般的なテキストの場合
  - `node.textContent = "Hello!!! "; // テキストを変更`
  - `node.innerHTML = "<b>Hello!!!</b>"; // HTMLを変更`
  - `node.style.background = "red"; // スタイルシートの背景色を変更`
- 画像の場合
  - `node.src = "http://snakamura.org/nakamura.jpg"; // 画像を変更`
  - `node.style.width = "400px"; // 画像の横幅を変更`

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

さらに

- テキストの内容を差し替えてみよう！
  - 要素のinnerHTMLに値を代入することで内容を差し替えることが可能

```
document.querySelector( “h1” ).innerHTML = “hogehoge”;
```

  - クリックされて内容を差し替えるには？

```
document.querySelector( “h1” ).onclick = function(){
  this.innerHTML = “hogehoge”;
}
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演習

- 「プログラミング演習II」という部分をクリックすると，「CMP2」と置き換えるようにせよ

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

DOMを取得してイベント追加

- ボタンのIDを取得して，クリックされたら「やぁ」というメッセージを表示する

```
<script>
  window.onload = function(){
      var button = document.getElementById("btn");
      button.onclick = function(){
          alert( "やぁ！" );
      }
  }
</script>

<input type="button" id="btn">
```

document.querySelector("#btn"); でもOK

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

メッセージを変更してみよう

- クリックするとメッセージを変更！

```
<html><head>
<script>
window.onload = function(){
  var msg_tag = document.querySelector( "#msg" );
  msg_tag.onclick = function(){
    this.innerHTML = "こんばんは";
  }
}
</script>
</head>
<body>
<span id="msg">こんにちは</span>中村さん
</body>
</html>
```

this はそれ自体。下記でもOK。
msg_tag.innerHTML = "こんばんは";

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

占いを作ってみよう

- クリックするとメッセージを変更！

```
<html><head>
<script>
var uranai = ["大吉", "中吉", "吉", "大凶"];
window.onload = function(){
  var result_tag = document.querySelector( "#msg" );
  result_tag.onclick = function(){
    var rand = Math.floor( Math.random() * 4 );
    this.innerHTML = uranai[ rand ];
  }
}
</script>
</head>
<body>
あなたの運勢は<span id="msg">？？</span>です
</body>
</html>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

よく使うので…

- 0以上1より小さい値をランダムに取得

```
Math.random();
```

- valueの小数点以下を切り捨てる

```
Math.floor( value );
```

- 0, 1, 2, 3の値をランダムに取得

```
Math.floor( Math.random() * 4 );
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

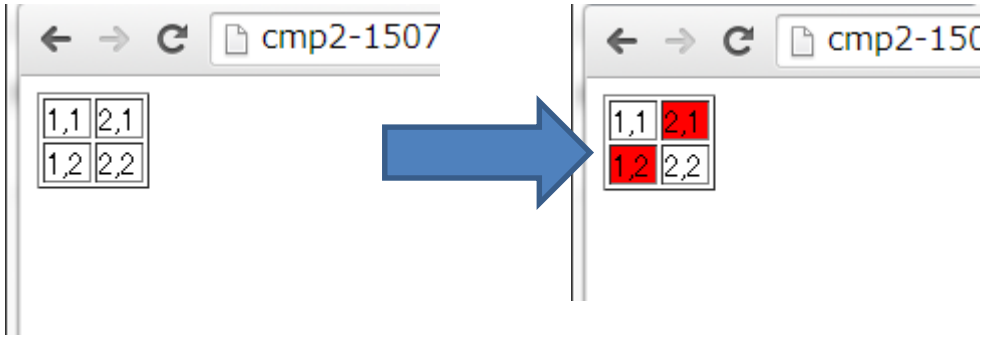
演習

- 診断メーカーのようなものを作ってみましょう
  - 「誰がいつどこで何をどうした」を作ってみよう
    - XXがXXでXXをXX
    - XXの部分をクリックするとそこの結果が変わるようにしてみましょう
  - 5連ガチャっぽいものを作ってみよう
  - 実行結果に応じて画像を切り替えてみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

表でクリックされた場所を赤色に

- 2x2の表を作成し，その表の中でクリックされた場所を赤色にするプログラムの作成

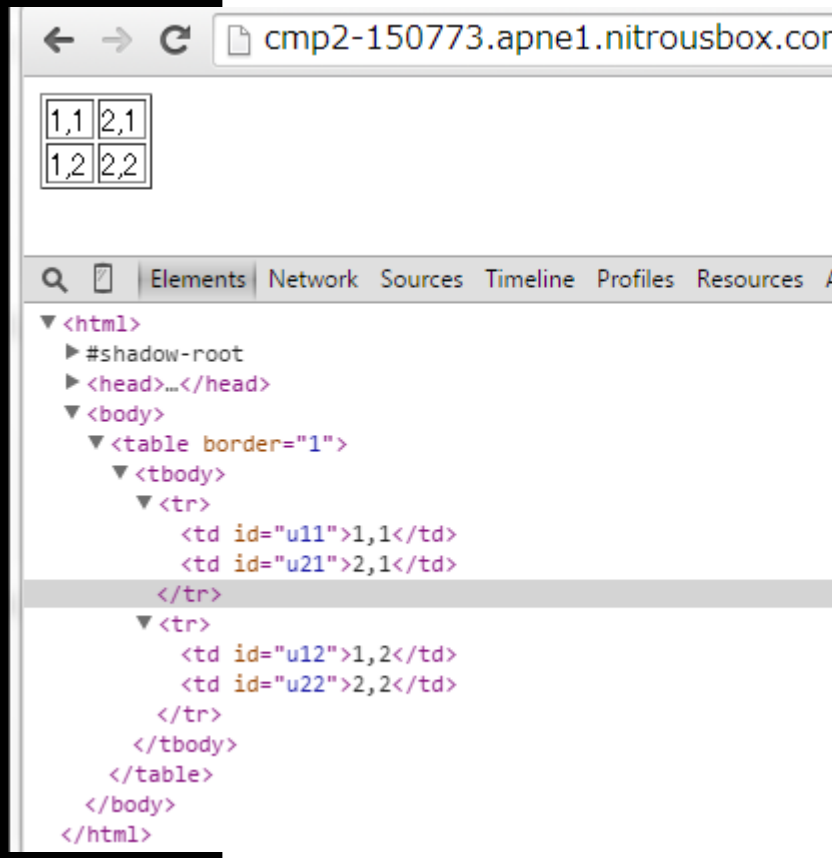


- ヒント
  - table, tr, tdタグを利用して2x2の表を作成
  - そのそれぞれにIDを付与
  - それぞれのクリックイベントに関数を割り当て
  - 割り当てた関数で背景色を変更

document.querySelector("#u11"); でもOK

```
<script>
  window.onload = function(){
    var td11 = document.getElementById("u11");
    td11.onclick = function(){
      this.style.background = "red";
    }
    var td12 = document.getElementById("u12");
    td12.onclick = function(){
      this.style.background = "red";
    }
    var td21 = document.getElementById("u21");
    td21.onclick = function(){
      this.style.background = "red";
    }
    var td22 = document.getElementById("u22");
    td22.onclick = function(){
      this.style.background = "red";
    }
  }
</script>

<table border=1>
  <tr><td id="u11">1,1</td><td id="u21">2,1</td></tr>
  <tr><td id="u12">1,2</td><td id="u22">2,2</td></tr>
</table>
```



```
<script>
    window.onload = function(){
      var tdunit = new Array( 2 );
      for( var x=0; x<2; x++ ){
        tdunit[x] = new Array( 2 );
      }
      for( var x=0; x<2; x++ ){
        for( var y=0; y<2; y++ ){
          tdunit[x][y] = document.getElementById("u"+(x+1)+(y+1));
          tdunit[x][y].onclick = function(){
            this.style.background = "red";
          }
        }
      }
    }
</script>

<table border=1>
      <tr><td id="u11">1,1</td><td id="u21">2,1</td></tr>
      <tr><td id="u12">1,2</td><td id="u22">2,2</td></tr>
</table>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

複数取得する場合は？

- 下記のメソッドの場合結果を複数取得
  - getElementsByClassName, getElementsByTagName, document.querySelectorAll
  - 結果は配列として取得できるため，下記のように処理することが可能

```
<script>
    window.onload = function(){
        var elems = document.getElementsByTag("td");
        for( var i=0; i<elems.length; i++ ){
            elems[i].style.background = "red";
        }
    }
</script>
```

document.querySelectorAll("td"); でもOK

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

配列を作らないでも...

```
<script>
   window.onload = function(){
      var units = document.getElementsByTagName("td");
      for( var i=0; i<units.length; i++ ){
         units[i].onclick = function(){
            this.style.background = "red";
         }
      }
   }
</script>

<table border=1>
      <tr><td id="u11">1,1</td><td id="u21">2,1</td></tr>
      <tr><td id="u12">1,2</td><td id="u22">2,2</td></tr>
</table>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

DOMの情報を取得する

- 他の実装方法
  - 引数としてthis（自分自身）を渡す
  - ここでthisはそのタグ（ノード）自体になるので node.innerHTML = "ひげひげ"; とできる

```
<script>
function change( node ){
    node.innerHTML = "ひげひげ";
}
</script>

<div id="hoge" onclick="change( this )">ほげほげ
</div>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

window.onload って何？

- ウェブページがロードされたら～という意味
  - この中にウェブページがロードされてからのプログラムを追加しよう！（ロードされる前に実行してしまうとおかしくなることがあるので要注意！）

```
<script>
  window.onload = function(){
    // ここにプログラムを書く！
  }
</script>
```

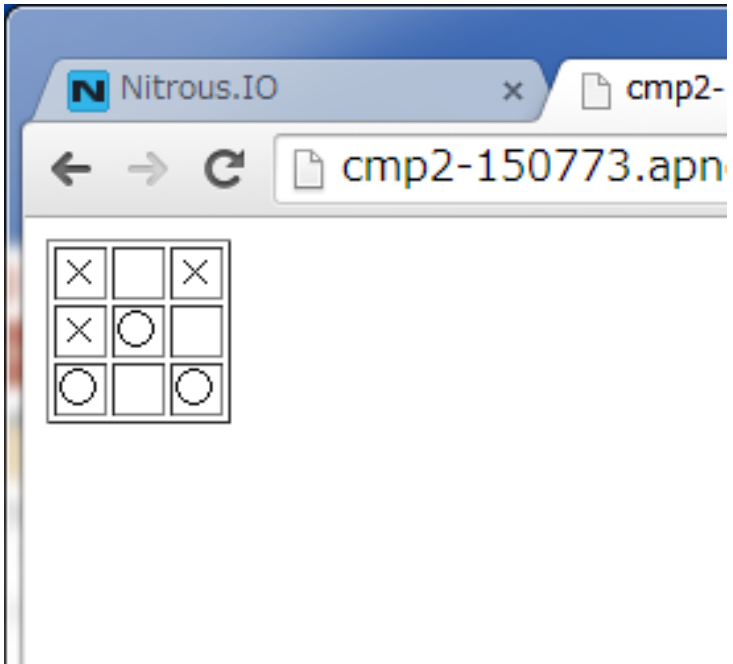
明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演習

- 2x2のテーブルのプログラムを拡張して，3x3の表の色が変わるページを作成してみよう！

- 3x3のテーブルのそれぞれの中にある「○」「×」「　」をクリックする度に
  - 「○」から「×」
  - 「×」から「　」
  - 「　」から「○」

  へと変化するようにせよ



明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

時間情報を取得する

- Date() というオブジェクトで取得可能

```
<script>
window.onload = function(){
  var dateObj = new Ðate();
  var time_tag = document.getElementById( "time" );
  time_tag.innerHTML = dateObje.getTime();
}
</script>

<div id="time">ほげほげ</div>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

定期的に実行

- setInterval(function()
  { 実行する内容 }, ミリ秒);
  - 指定ミリ秒後に指定の操作を実行する

表でクリックされた場所を赤色に下記プログラムを追加！

```
setInterval(function(){
    if( td11.textContent === "○" ){
        td11.textContent = "";
    } else {
        td11.textContent = "○";
    }
},1000);
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

時間情報を取得する

- Date() というオブジェクトで取得可能

```
<script>
window.onload = function(){
  setInterval( function(){
    var dateObj = new Ðate();
    var time_tag = document.getElementById( "time" );
    time_tag.innerHTML = dateObje.getTime();
  }, 100 );
</script>

<div id="time">ほげほげ</div>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

時分秒を取得する

- Date() というオブジェクトで取得可能

```
<script>
window.onload = function(){
  setInterval( function(){
    var dateObj = new Date();
    var time_tag = document.getElementById( "time" );
    time_tag.innerHTML
       = dateObj.getHours()
       + ":" + dateObj.getMinutes()
       + ":" + dateObj.getSeconds();
  }, 100 );
</script>
<div id="time">ほげほげ</div>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

画像を定期的に切り替える

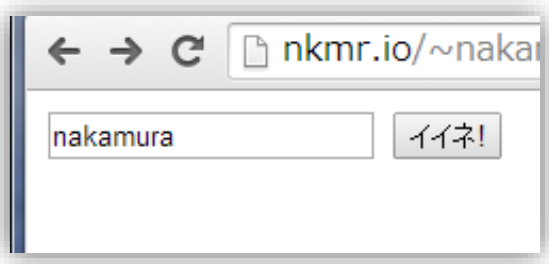
- 1秒おきに画像をランダムに変更

```
<html><head>
<script>
var imgs = ["apple.jpg", "orange.jpg", "fms.jpg"];
window.onload = function(){
  var img_tag = document.getElementById( "slide" );
  setInterval( function(){
    var rand = Math.floor( Math.random() * 3 );
    img_tag.src = imgs[ rand ];
  }, 1000 );
}
</script>
</head>
<body>
<img src="nakamura.jpg" id="slide">
</body>
</html>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演習

- テキスト入力ボックスと送信ボタンを用意し，入力ボックスで入力されている内容に応じて送信ボックスの状態を切り替えてみよう
- ここでは，入力ボックスに自分の名前が入力されていたら，送信ボックスのボタンを「イイネ」と変更して，送信可能にせよ



明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

手順（ヒント）

- フォームタグやテキストボックス，送信ボックスを用意し，それぞれにIDを付与する
- 送信ボックスを最初利用不可にするため disabled に設定しておく
- テキストボックスで何か文字が入力されていたら入力内容をチェックし，何らかの文字が入力されていたら送信ボックスを押せるようにせよ（XXX.disabled = false）
- 入力内容が自分の名前だったらテキストを変更する
- テキストボックスへの入力は onkeydown で取得できるのでそのイベント（関数）を追加しよう！

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

JavaScriptは別ファイルへ

- HTMLとJavaScriptが混在しているとぐちゃぐちゃになってしまう（PHPとJavaScriptが混在すると恐ろしいことに）
- 可能な限りJavaScriptファイルは別ファイルへ

- main.js などに保存し script タグで呼び出し
  - src で呼び出すファイルのパスを指定する

```
<script src="main.js"></script>
```

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

JavaScriptの良いところ

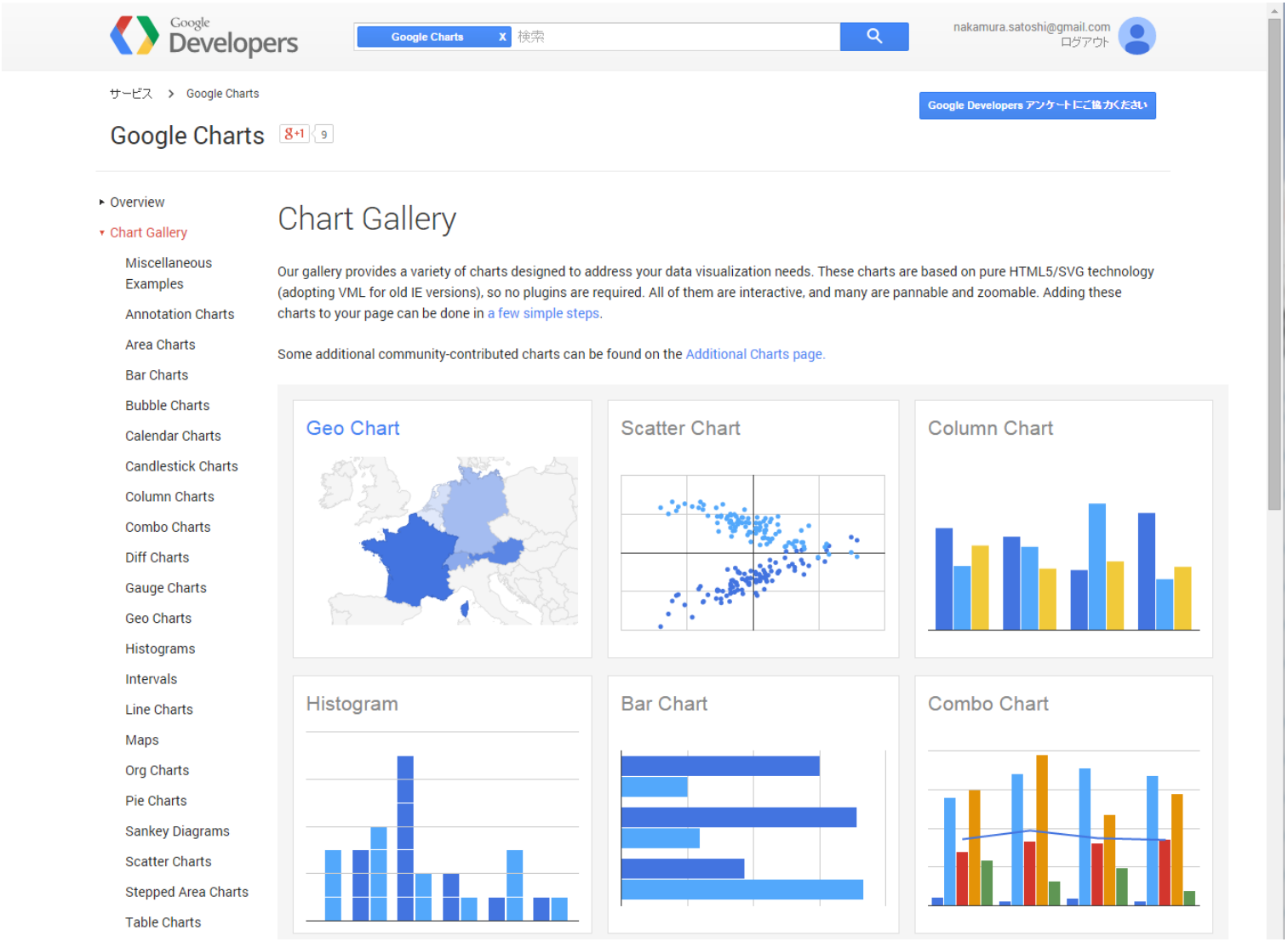
- 開発者がかなり多いため，ライブラリも豊富
  - グラフを描画する（GeoChart）
  - 地図を表示する
  - Processingのプログラムを実行する
  - 色々便利にするためのフレームワークもある！
    - JavaScriptを一から全部書いていくのはかなり面倒

```
<script src="呼び出すJSのURL"></script>
```

https://developers.google.com/chart/interactive/docs/gallery?hl=ja

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

Google Charts



明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

演習

- 総合数理学部 学生数のパイチャートを作成
  - 現象数理 80人
  - FMS 100人
  - ND 80人
- 日本の各都道府県の人口毎の地図を作成
  - Tokyo 13286735人
  - Kyoto 2620210人
  - Fukuoka 5090712人
- Speed メーターを作成せよ

Pie Chart

```
<html>
 <head>
 <script type="text/javascript" src="https://www.google.com/jsapi"></script>
 <script type="text/javascript">
  google.load('visualization', '1.0', {'packages':['corechart']});
  google.setOnLoadCallback(drawChart);

  function drawChart() {
    var data = new google.visualization.ĐataTable();
    data.addColumn('string', 'Name');
    data.addColumn('number', '人数');
    data.addRows([['現象数理', 80], ['FMS', 100], ['NĐ', 80]]);

    var options = {'title':'学生数比較', 'width':400, 'height':300};
    var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
        chart.draw(data, options);
  }
 </script>
 </head>

 <body>
   <div id="chart_div"></div>
 </body>
</html>
```

Geo Chart

```
<html>
 <head>
  <script src='https://www.google.com/jsapi'></script>
  <script>
    google.load('visualization', '1', {'packages': ['geochart']});
    google.setOnLoadCallback(drawMarkersMap);
    function drawMarkersMap() {
    var data = google.visualization.arrayToDataTable([
       ['県', '人口'], ['Tokyo', 13286735],  ['Kyoto', 2620210],  ['Fukuoka', 5090712]
    ]);
    var options = {
       region: 'JP',
       displayMode: 'markers',
       colorAxis: {colors: ['green', 'blue']}
    };
    var chart = new google.visualization.GeoChart(document.getElementById('chart_div'));
    chart.draw(data, options);
   };
  </script>
  </head>
  <body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
  </body>
</html>
```

https://google-developers.appspot.com/chart/interactive/docs/gallery/geochart

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

Processing.js / p5.js

- Processing.js や p5.js はProcessingのプログラムを実行するためのライブラリ

明治大学総合数理学部
先端メディアサイエンス学科
中村聡史研究室

課題

- 組番号のフォルダ内に課題レポートのトップページをHTMLで作成せよ
  - タイトルと名前、学籍番号などは含むようにせよ
  - リンク先に各レポート課題のページを作成せよ
  - リンク元でどういう課題を作ったかを軽く説明せよ

- 課題
  1. JSを利用し，ユーザが入力した名前込みで何らかの結果を返す，診断メーカーのようなもの（クリックなどで結果が組み合わせで変わるもの）を作成せよ
  2. JSのsetIntervalを用いて時計のようなものを作成せよ．画像を使うなど見栄えを工夫してください
  3. Processing.js または p5.js を利用して，自分がこれまでに作ったProcessingのプログラムを動かせ