WEBプログラミング/演習
第3回

スケジュール

- 第1回：Webプログラミングの基本
- 第2回：JavaScriptの基本
- 第3回：オブジェクト
- 第4回：関数
- 第5回：オブジェクト指向
- 第6回：クライアントサイドJavaScript
- 第7回：公開APIの利用(1)
- 第8回：公開APIの利用(2)
- 第9回：Pythonプログラミングの基礎(1)
- 第10回：　Pythonプログラミングの基礎(2)
- 第11回：サーバサイドプログラミング(1)
- 第12回：サーバサイドプログラミング(2)
- 第13回：サーバサイドプログラミング(3)
- 第14回：学習内容の振り返り

この講義について（再記）

- 概要
  - オブジェクト指向スクリプト言語JavaScriptは、Webブラウザ、IoT機器のプログラミング言語としてまた、Pythonは、科学計算やWebサーバのプログラミング言語として広く使われています。
  - 本講義の前半ではWebブラウザで動作するプログラム（JavaScript）、後半ではサーバー上で動作するプログラム（python）を学び、Webアプリケーションが動作する仕組みを理解します。
- 成績評価
  - レポート課題（ほぼ毎回の授業で出題）と最終レポート課題で評価します。
  - 定期試験期間には試験は行いません．小課題と最終レポートの評価の割合は，4:6とします．
  - A+，A，B，C，D，Fの6段階評価を行い，D以上を合格とします．特別な理由なく全体の出席日数が3分の2に満たない場合は不合格となります．

注意事項

- シラバスにある通りWebプログラミングとWebプログラミング演習は同時に履修を行ってください.
- 講義資料，課題はWebプログラミング/演習で共通です．2コマかけて取り組むことを想定しています.
- 出席は教室の出席登録リーダーにて．3,4限それぞれ登録してください.
- Course Powerの「Webプログラミング演習」は基本的に利用しないつもりです．連絡や資料提供は「Webプログラミング」の方で行います.

オブジェクト

CoursePower（第3回>プログラム一式）からプログラム一式をダウンロードください．これらはexercise3（フォルダ名）として扱いましょう．

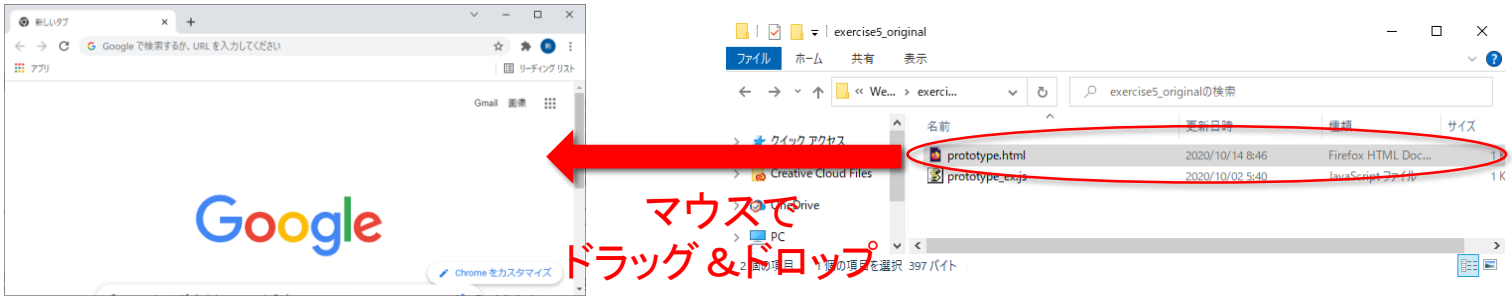
[復習] console.logを確かめよう

- JavaScriptの簡易的なデバッグでは，わざわざサーバ上で動かす必要はない
  - ローカル（自分のノートPC）のWebブラウザで，javascriptを伴うhtmlを開いて実行させよう（次頁の図も参考に）
    - 手っ取り早いのは，
      - Google Chromeを立ち上げる.
      - 編集したhtmlファイル（今回だとobject.html）をエクスプローラーなどからChromeへドラッグする.
  - exercise3のファイルを同じフォルダに入れておくこと．前回までのファイルは一緒に入れないこと.

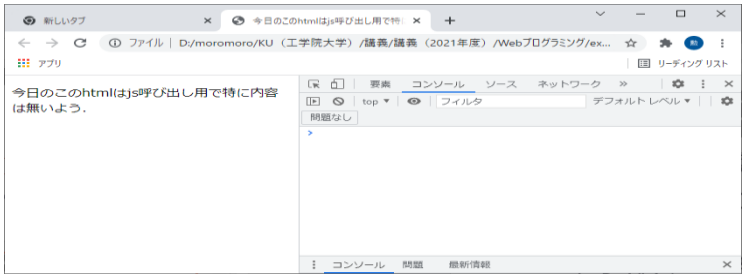
- console.log(hogehoge)は，Chrome Developer Tools の 「Console」タブのところに出力される
  - 開いて結果を確かめよう（使い方はCoursePowerにあるFAQファイルを参考に）

今日の課題について

- 今日もGCPは使いません.
- ローカルの同じフォルダに今日DLしたファイル全てを置いて，htmlファイルをブラウザで開く.
  1. ブラウザ（Google Chromeなど）を起動して，CoursePowerからダウンロードした演習用HTMLファイルを，ブラウザ上にドラッグ＆ドロップする.



  2. あとはいつもどおりデベロッパーツールを開くだけ（F12を押す，Chrome上で右クリック->「検証」を選択，等）.



オブジェクトって何？

- オブジェクトは概念を表す総合体と考えます
  - 代表的なオブジェクトには，Array（配列），Date（日付），String（文字列）などがあります
- オブジェクトはざっくり言うと，プロパティ + メソッドです
  - プロパティは，オブジェクトの内容を表現するための要素です
    - Dateオブジェクトであれば，年，月，日，時，分，秒など日付を表現するための属性がプロパティです.
  - メソッドは，オブジェクトの振る舞いを表現するためのものです.
    - Dateオブジェクトであれば，日付の加算（9月30日に1日足したら，10月1日なるなど）や，曜日を得るなどの振る舞いが定義されています

オブジェクトを生成する

- new 演算子でオブジェクトの生成を行います

  let d = new Date();
  - これで，日時オブジェクトが生成されます
  - 生成されたオブジェクトをインスタンスと言います

- オブジェクトとインスタンスの関係
  - オブジェクトは唯一無二の概念
    - Dateオブジェクト = 日時という概念
  - インスタンスは具体的な値を伴う実体（中身みたいな）
    - Dateインスタンス = 10月1日などの値を伴う実体
  - 鯛焼きの鉄板（金型）⇔　鯛焼き（しろあん，つぶあん，さくらあん，，，）　あくまでもイメージね.

- オブジェクトは基本的にはインスタンスを作成して，インスタンスを操作する

組み込みオブジェクト

- 組み込みオブジェクトとは，JavaScriptで標準的に用意されたオブジェクトで，特別な操作なく使うことができます.

- 代表的な組み込みオブジェクト
  - 配列：Arrayオブジェクト
  - 文字列：Stringオブジェクト
  - 数値：Numberオブジェクト
  - 真偽値：Booleanオブジェクト
  - 日時：Dateオブジェクト
  - 正規表現：RegExpオブジェクト

- 今日は，これらのオブジェクトの使い方を学びます.

基本データ型とオブジェクト

- 配列や文字列，数値は基本データ型です．

- 基本データ型もオブジェクトなのですが，インスタンス作成のための記述が省略できます．
  - 基本データ型？忘れた人は->第2回スライド25ページ

let s = “こんにちは”;
これは
let s = new String(“こんにちは”);
と同じことです．

プロパティとメソッドの呼び出し

- オブジェクトはプロパティとメソッドをもっています．以下のように”.”でつなげて呼び出します．
  - 変数名.プロパティ
  - 変数名.メソッド()

- プロパティの呼び出し
  - 文字列オブジェクトはその長さ（length）をプロパティとしてもっています．

```
let s = “こんにちは”;
s.length; // 5が入っている
```

- メソッドの呼び出し
  - 文字列オブジェクトは探したい文字列の位置を返すメソッド（indexOf）をもっています．

```
let s = “こんにちは”;
s.indexOf(“ち”); // 3が返ってくる
```

Stringオブジェクトによる文字列の操作

- 主要なプロパティとメソッド

indexOf(substr)	文字列前方から部分文字列substrを検索し，その位置を返す
lastIndexOf(substr)	文字列後方から部分文字列substrを検索し，その位置を返す
slice(start,end)	文字列からstart+1からend文字目を抽出
substr(start,cnt)	文字列のstart+1文字目からcnt文字分抽出
split(str)	文字列を分割文字列strで分割し，その結果を配列として取得
toLowerCase()	小文字に変換
concat(str)	文字列の後ろに文字列strを結合
trim()	文字列の前後から空白を削除
length	文字列の長さ

[課題1] 色々文字列操作しよう

- object_ex.jsに以下を出力するようなプログラムを書こう
  - 出力は，console.logを使って，Developer Toolのconsoleタグで確認すればよい
  - スライド13ページの情報だけでは足りないので適宜調べること

- let s = “にわにはにわにわとりがいる”;
- 上記の文字列に対し，以下の情報を得よ．ただし先頭は0文字目とする.
  - “と”は先頭から何文字目にあるか？
  - この文字列の長さは？
  - “わ”で文字列を分割せよ
  - “にわ”は何回出現したか？
  - “へ”は先頭から何文字目にあるか？
  - 先頭3文字目から後方4文字目までを得よ
  - “かもしれない”を後方に追加せよ
  - 7文字目の文字を得よ

Numberオブジェクトによる数値の操作

- 主要なプロパティとメソッド

MAX_VALUE *	Numberオブジェクトが扱える最大の数
MIN_VALUE *	Numberオブジェクトが扱える最小の数
NaN *	非数値
POSITIVE_INFINITY *	正の無限大
toString(rad)	rad進数の値に変換（toString()だとどうなる？）
toFixed(dec)	小数第dec位になるように四捨五入
isNaN(num) *	NaNであるかと判定

静的プロパティ/メソッド

- スライド12では，プロパティやメソッドは変数，つまりインスタンスから呼び出すという形で説明しました

- オブジェクトそのものがプロパティやメソッドを持つケースがあります
  - スライド15の * がついているもので，以下で呼び出す
  - オブジェクト名.プロパティ
  - オブジェクト名.メソッド()

- 例えば
  - MAX_VALUEは Number.MAX_VALUE
  - inNaN() は Number.isNaN(5) // falseになる

[課題2] 色々数値操作しよう

- object_ex.jsに以下を出力するようなプログラムを書こう
  - スライド14の情報だけでは足りないので適宜調べること
- let n1 = 255; に対し以下の情報を得よ
  - 16進数に変換せよ
  - 8進数に変換せよ
- let n2 = 123.45678; に対し，以下の情報を得よ
  - 小数点3桁で四捨五入せよ
  - 小数点7桁で四捨五入せよ
  - 指数表記を得よ

Mathオブジェクトによる数学演算

- 主要なプロパティとメソッド

abs(num)	numの絶対値を得る
max(num1,num2,…)	num1,num2,…の最大値を得る
min(num1,num2,…)	num1,num2,…の最小値を得る
pow(base,p)	べき乗をえる（baseのp乗）
random()	0~1の乱数を得る
round(num)	四捨五入
sqrt(num)	平方根を得る
PI	円周率
cos(num)	コサイン
sin(num)	サイン
E	自然対数の底
log(num)	自然対数

Mathオブジェクトのインスタンス

- Mathオブジェクトは数学演算という「概念」のオブジェクトなので，インスタンスを持ちません.

  let m = new Math();

  はエラーとなり実行できません.

- すべて，静的プロパティ/メソッドとして呼び出すことになります.
  - 前回までに出てきた Math.rondom() も静的メソッド呼び出しだったわけです.

[課題3] 色々数学演算しよう

- object_ex.jsに以下を出力するようにプログラムを書け
  - スライド18の情報だけでは足りないので適宜調べること

- 以下の変数に対して次の処理をせよ
  - let n4 = -255;
  - let n5 = 123.45678;
  - let n6 = 81;
- n4の絶対値を得よ
- n4,n5,n6の最大値を得よ
- n5を四捨五入せよ
- n6の平方根を得よ
- n6の自然対数を得よ
- n4の2乗を得よ
- n5 - 円周率を求めよ
  （円周率はMathオブジェクトを使って求めること）

Arrayオブジェクトによる配列操作

- 主要なプロパティとメソッド

length	配列の長さを得る
toString()	[要素,要素,…]という形式で文字列に変換
indexOf(elm)	指定した要素elmに合致した最初の要素の添字を返す
concat(ary)	指定配列aryを現在の配列に連結
join(del)	配列内の要素を区切り文字delで連結
slice(start,end)	start+1からend番目の要素の抜き出し
pop() *	配列末尾の要素を取得し，削除
push(data) *	配列末尾に要素dataを追加
shift() *	配列先頭の要素を取得し，削除
reverse() *	逆順に並べ替え
sort() *	配列を昇順に並べ替え

破壊的メソッド

- そのメソッドによって，元の配列のデータが変化するものを破壊的メソッドと呼びます
  - スライド21 の * のメソッド

```
let a = [3,2,1];
a.sort();
```

  - この結果，aには[1,2,3]が格納されていることになる

[課題4] 色々配列操作しよう

- object_ex.jsに以下を出力するようにプログラムを書け
  - スライド21の情報だけでは足りないので適宜調べること

- 以下の変数に対して次の処理をせよ．ただし先頭は0番目（0要素目）とする．
  - let a1 = [‘Sato’,’Takae’,’Osada’,’Hio’,’Saitoh’,’Sato’];
  - let a2 = [‘Yabuki’,’Aoki’,’Moriyama’,’Yamada’];
- a1の長さを得よ
- a1を文字列に変換せよ
- a1にa2を連結せよ
- a1を’-’で連結して文字列にせよ
- a1の2要素目から4要素目までを得よ
- a1の末尾にa2の2番目の要素を追加せよ
- a1を降順に並べよ

Mapオブジェクトによる連想配列操作

- 主要なプロパティとメソッド

size	要素数を得る（集合概念なのでlengthではない）
set(key,val)	キーkey/値valペアの要素を追加
get(key)	指定したキーkeyの要素を取得
has(key)	指定したキーkeyが存在するかを判定
delete(key)	指定したキーkeyの要素を削除
clear()	すべての要素を削除
keys()	すべてのキーを取得
values()	すべての値を取得
entries()	すべてのキー/値を取得

そもそも連想配列とは？

- 配列は添字と値の構造だった
  - let a = [‘たこ焼き’, ’味噌カツ’, ‘ちゃんぽん’];
  - 0番目（添字）に’たこ焼き’という値が対応する
  - 1番目（添字）に’味噌カツ’という値が対応する

- 連想配列は添字に変わって「key（任意の型）」が扱えるもの
  - let m = new Map();
  - m.set(‘大阪’, ’たこ焼き’);
  - m.set(‘名古屋’, ’味噌カツ’);
  - ‘大阪’（key）に’たこ焼き’という値が対応する

[課題5] 色々連想配列操作しよう

- object_ex.jsに以下を出力するようにプログラムを書け
  - スライド24の情報だけでは足りないので適宜調べること

key	value
dog	わんわん
cat	にゃー
mouse	ちゅー

- 次の処理をせよ
  - 右の表のkeyとvalueを連想配列として，変数mに格納せよ
  - mにduckのkeyが存在するか確認し，存在すれば値を取得せよ，存在しなければ，duckの値として”がーがー”を格納せよ.
  - mよりmouseのkeyを削除せよ
  - mの要素数を得よ
  - mのすべての値を取得し，結合して１つの文字列にせよ

Setオブジェクトによる集合操作

- 主要なプロパティとメソッド

size	要素数を得る（集合概念なのでlengthではない）
add(val)	指定した値valを追加
has(val)	指定した値valが存在するかを判定
delete(val)	指定した値valを削除
clear()	すべての要素を削除
values()	すべての値を取得

- Mapオブジェクトのvalue部分だけ（概念的にはkey部分だけと思うほうが近い）と思えばよいです．純粋に値の集合を扱います．
  - 集合なので，重複する値は存在できません．

Dateオブジェクトによる日時操作

- 主要なプロパティとメソッド（*は静的）

getMonth()	月を取得，同様に年，日，曜日の取得メソッドがある
getMinutes()	分を取得，同様に時間，秒，ミリ秒の取得メソッドがある
getTime()	いわゆるUNIXタイム（ミリ秒）が得られる
setDate(d)	日dを設定，同様に年，月の設定メソッドがある
setSeconds(s)	秒sを設定，同様に時間，分，ミリ秒の設定メソッドがある
parse(dat) *	日付文字列datを解析し，そのUNIXタイムを取得
now() *	協定世界時での現在の日時をUNIXタイムで取得
toLocaleString() *	ローカル時を文字列で取得

Dateオブジェクトのコンストラクタ

- コンストラクタはオブジェクト生成時に使うオブジェクトと同名のメソッドです
  - let d = new Date();
    のDate()がコンストラクタです
  - 引数にオブジェクトの初期化パラメータを渡します

Date()	引数に何も指定しなければ, 現在の日時が入ります
Date('2016/12/04 20:07:15')	文字列で日時を指定して生成することもできます
Date(2016,11,4,20,07,15,500)	日付を各値として指定します．なお，月は0~11です.
Date(1480849635500)	UNIXタイムを指定することも可能です

日時の加算/減算

- 直接的に加算減算をするメソッドは用意されていません．set系で値を設定するときに元の値に対して加減算することになります．

```
let d = new Date();
d.setMonth(d.getMonth() + 1);
```

  - これで，dには現在時刻の1ヶ月後の日時が入ることになります．
  - この時，範囲を超える演算であっても大丈夫です．12月のときに +1したなら，翌年の1月という扱いになります．

日時の差分

- 日付Aと日付Bが何日離れているのか？ということを計算したいときがあります.
  - 5日経過したらリマインダを投げるとか
  - 残念ながらJavaScriptでは簡単に計算できる方法がありません.
  - 両方のUNIXタイムを引き算して，日表現になおします.

```
let d1 = new Date(2017,4,15);
let d2 = new Date(2017,5,20);
let diff = (d2.getTime() – d1.getTime()) /
    (1000*60*60*24);
```

  - これで，diffには36という値が入ります.
  - (1000*60*60*24)で割ることで，ミリ秒を日付にしているわけです.

[課題6] 日付操作とストップウォッチ

- object_ex.jsに以下を出力するようにプログラムを書け
  - スライド28の情報だけでは足りないので適宜調べること
- 現在の日時を変数todayに格納し以下の情報を得よ
  - todayのミリ秒を得よ
  - todayに40日後を設定せよ
  - todayの秒数が3の倍数のときに「たこ焼き」，それ以外のときに「味噌カツ」と表示せよ
- for文で1から100000まで加算するのにかかった処理時間（ミリ秒）を表示せよ

RegExpオブジェクトによる正規表現

- そもそも正規表現って？
  - 文字列のパターンを表現する表記法です.
- ピカチュウの鳴き声を判定したいとします.
  - ピカチュウ，ピーカーチューウー，ピカピカー，ピカピーカ，などが代表的です.
  - 使われる文字列は限定的で，法則がありそうです
- パターン1：ピー*カー*チュー*ウー*
  - *は直前文字の0回以上の繰り返しです．このパターンでは以下を判定できます.
    - ピカチュウ，ピーカーチューウー，ピーーーカチュウ
- パターン2：　（ピー*カー*）+
  - ()でグループ化します．+は直前文字の1回以上の繰り返しです．このパターンでは，以下を判定できます.
    - ピカー，ピカピーカ，ピカピカピカピカー

JavaScriptで使える主な正規表現

パターン	意味
ABC	AとBとCが連続する文字列
[ABC]	AかBかCの1文字
[^ABC]	A,B,C以外の1文字
[A-Z]	AからZまでのいずれか1文字
A|B|C	AかBかC
X*	0回以上連続出現のX
X?	0または1回のX
X+	1回以上連続出現のX
X{n}	n回のX
X{n,}	n回以上連続出現のX
X{m,n}	m~n回連続出現のX
^	行の先頭
$	行の末尾
.	任意の1文字

[課題7] メールアドレスの判定

- @ns.kogakuin.ac.jpのアドレスや，@g.kogakuin.jpのアドレスを判定できる正規表現を考えよ.

- @より前はとり得る可能性のあるものを判定できること．ただし実在しないものを誤判定するのは仕方ないものとする.
  - 例えば，j399555は，現時点では存在しないが，将来的は存在し得る可能性があるので判定できる必要がある.
  - 一方，ggttss134は，まったく現在のルールに当てはまっていないためNGである.

正規表現オブジェクトの生成

- 他のオブジェクト同様にコンストラクタで生成します

  let 変数名 = new RegExp(‘正規表現’,’オプション’);

  - 引数は両方とも文字列であることに注意
  - 正規表現中で ¥ のように特殊な意味を持つものは ¥¥ としてエスケープする必要があります.
  - オプションは下表のものが使え，複数指定する場合は，‘gi’ のようにつなげて指定する.

g	文字列全体にマッチさせる（指定なしだと1つ見つかった時点で終える）
i	大文字/小文字を区別させる
m	複数行を個別に判定をさせる（先頭，末尾の扱いが変わる）
u	Unicode対応をする

正規表現による検索

let text = 私のアドレスはj399555@ns.kogakuin.ac.jpと
j399555@g.kogakuin.jpです．；
let reg = new RegExp(‘メールアドレスの正規表現’,’gi’);

- 上記に対して，

```
let results = text.match(reg);
for(let i = 0; i < results.length; i++){
        console.log(results[i])
}
```

で，正規表現にマッチした部分を表示できる
- matchは正規表現を引数に取るstringのメソッド
- メールアドレスの正規表現は課題7のものが入る

[課題8] メールアドレスの判定

- 実際に課題7で作成した正規表現を使って，スライド37を実行せよ.

- 例えば，let text = 私のアドレスは j399555@ns.kogakuin.ac.jpと j399555@g.kogakuin.jpです．；

としたら，

j39955@ns.kogakuin.ac.jp

j39955@g.kogakuin.jp

- が出力される.

[課題9] 色々使おう（自主課題）

- “令和は2019年5月1日からです．平成は1989年1月8日からです．昭和は1926年12月25日からです．大正は1912年7月30日からです．明治は1968年10月23日からです．”
- 上記の文章に対して，これまでに学んだことを駆使して，以下のような連想配列を生成せよ．
  - 考え方は色々あるので，各自工夫を凝らしてほしい．正規表現の検索機能を使わなくてもよい．sliceメソッドを駆使するとか．
  - 正規表現の検索機能だと，matchメソッド以外にもmatchAllメソッドを使う方法もある．

元号を格納する　　元号開始年月日（西暦）を格納する

key	value
令和	2019年5月1日
平成	1989年1月8日
…	…

成果物の提出（〆切：10/8）

- 作成および編集した以下のファイルを提出せよ．ただし課題9は自主課題とするので，回答は任意とする．
  - object_ex.js

- 提出はCoursePowerで行うこと