Webプログラミング Ⅱ

写真共有アプリの写真アップロード

3回目

山崎　大助

D.Yamazaki

アジェンダ

課題提出:2

VisualStudioCode準備

写真共有アプリの基本(FileAPI)
　アプリ仕様の説明(今日授業でおこなう場所)
　前回授業内容を少しだけ説明
　Fileボタン（JavaScript:Fileサイズチェック）

◇学べること
FileAPI
FileSize制限
JavaScript基礎（2年生で学んでいることの延長）

課題

課題

UXをUPするための課題

#select_btn

カメラ/写真選択

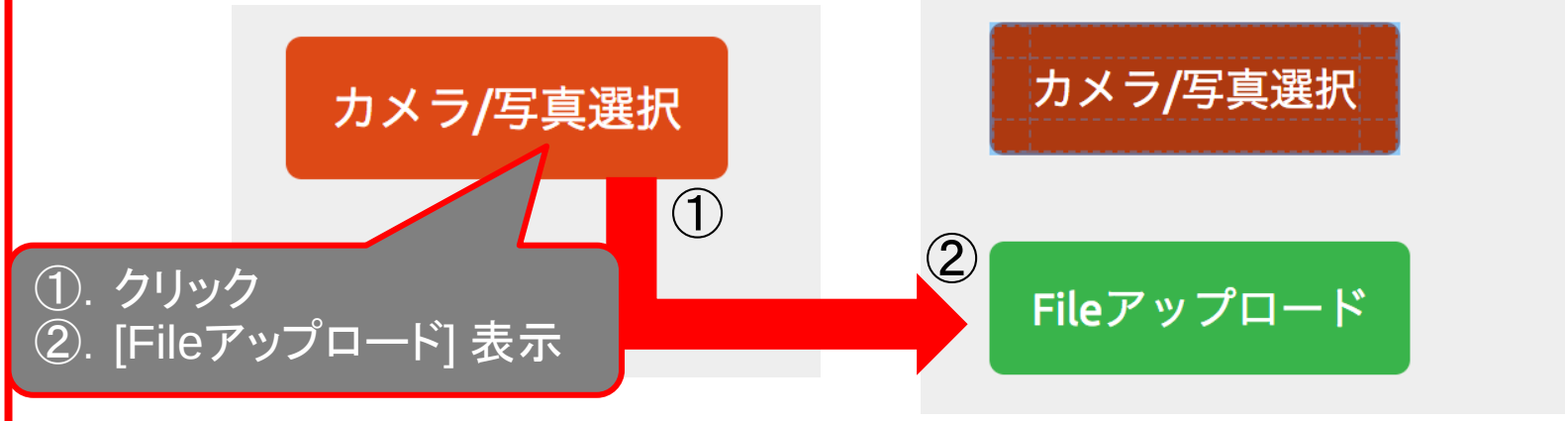
Fileアップロード

input[type="submit"]

1.「Fileアップロード」を最初は非表示にする

2.「カメラ/写真選択」で写真選択したら
「Fileアップロード」を表示する

動作の流れ

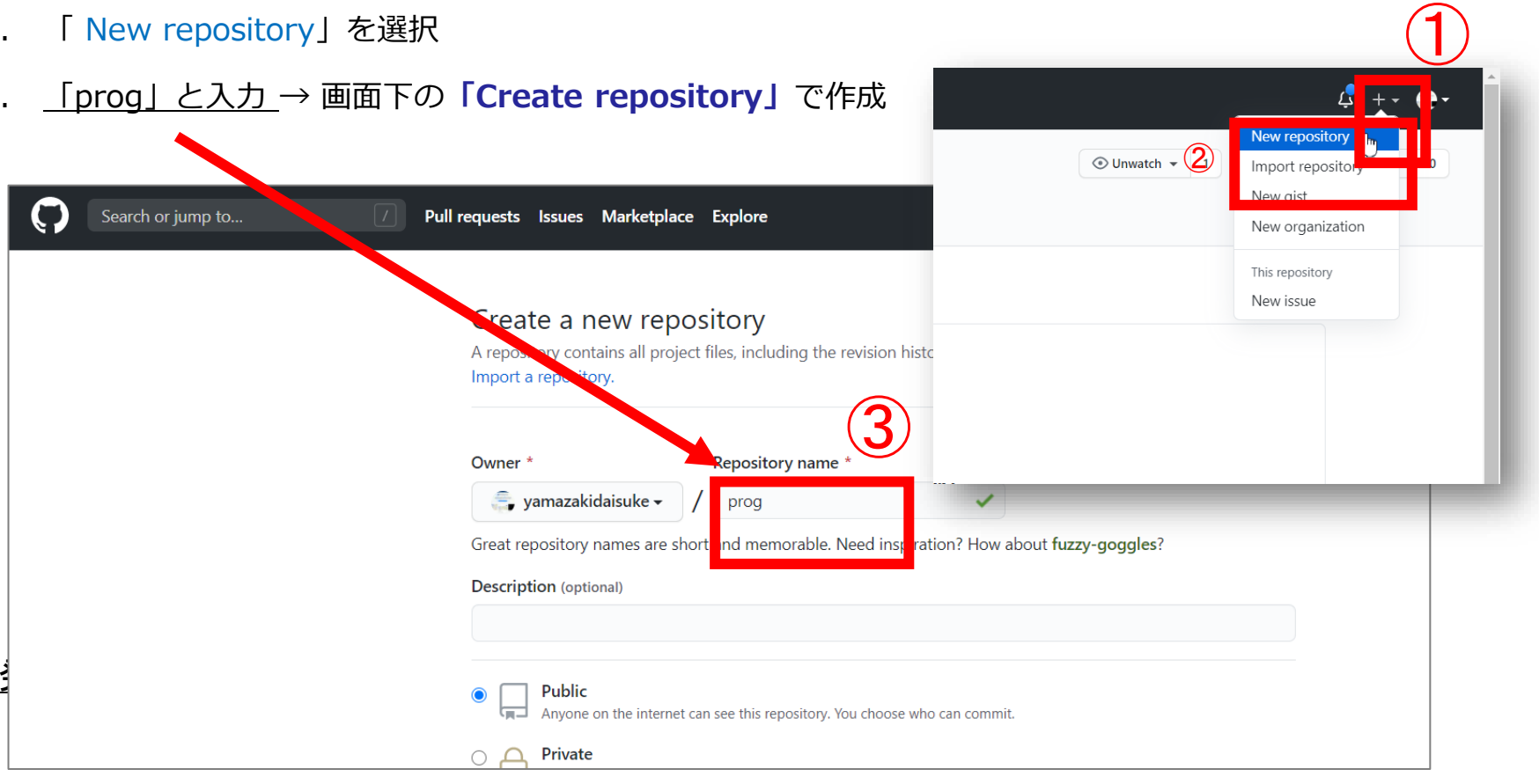


授業制作物：提出の方法1

1. 自分のパソコン上で「01」フォルダを作成
2. 「01」フォルダに課題に必要なファイル一式を入れる
3. Githubアカウントを作成
4. 登録解説ページを見ながらやりましょう
   https://shimapuku.com/development/github-account
5. 登録完了したら次のスライド

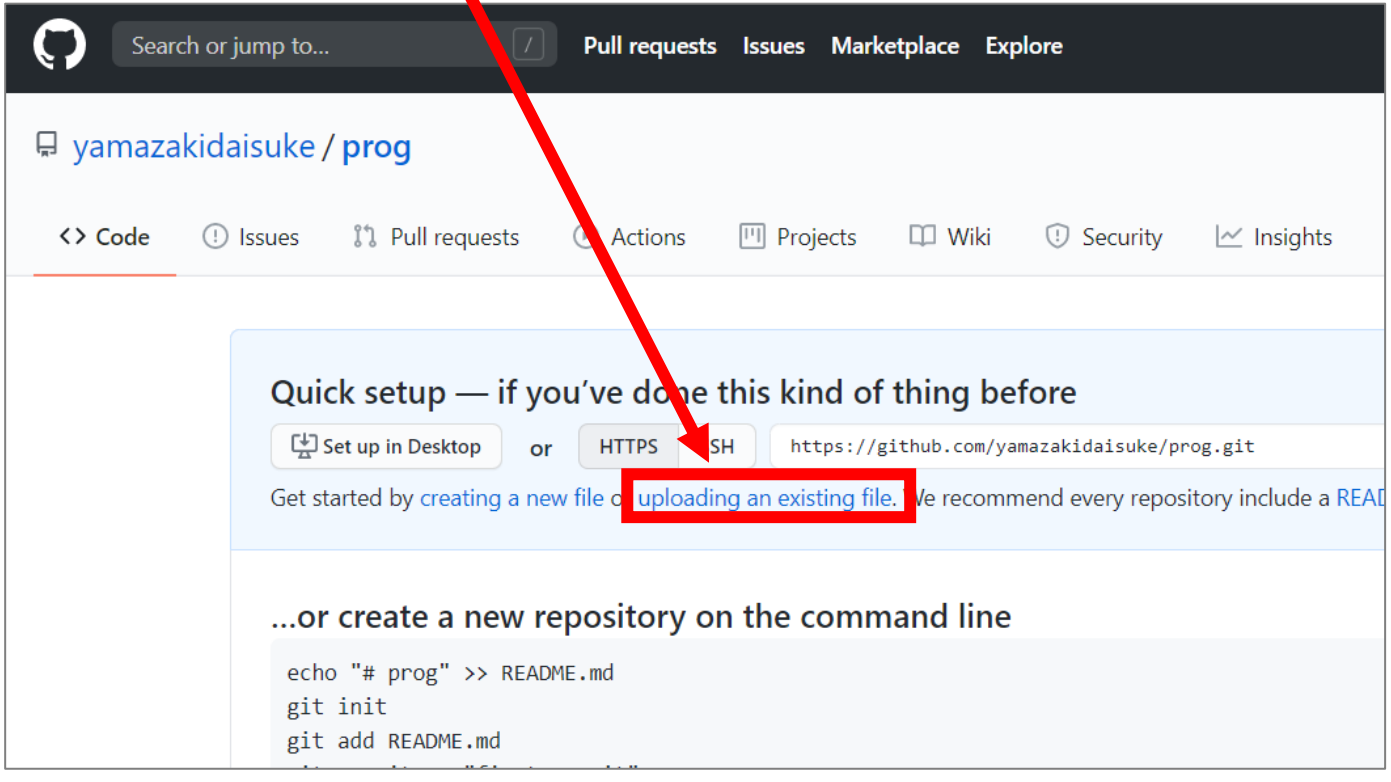
授業制作物：提出の方法2

1. 画面右上「＋」ボタンをクリック
2. 「 New repository」を選択
3. 「prog」と入力 → 画面下の**「Create repository」**で作成



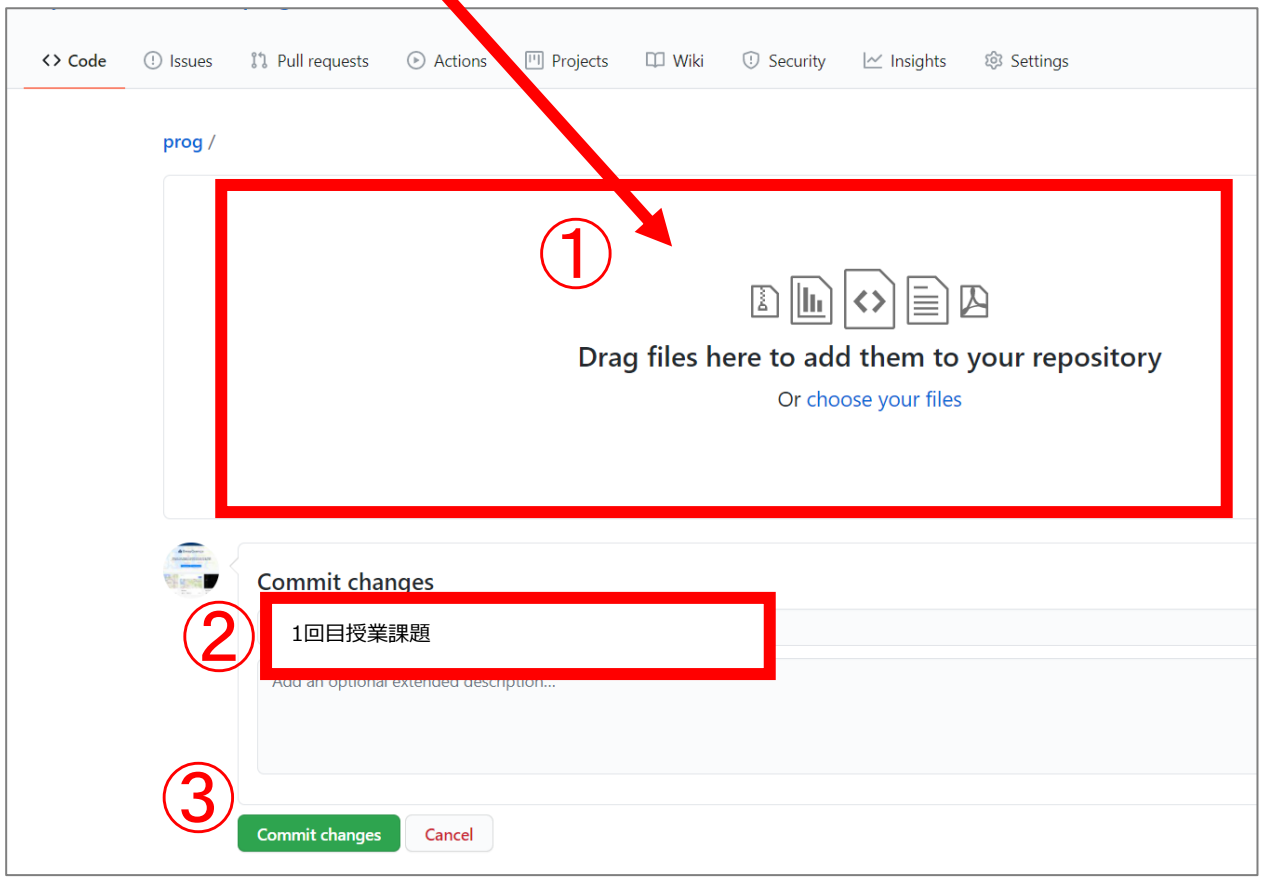
授業制作物：提出の方法3

1. progリポジトリが作成されました。
2. 「 uploading an existing file.」をクリック



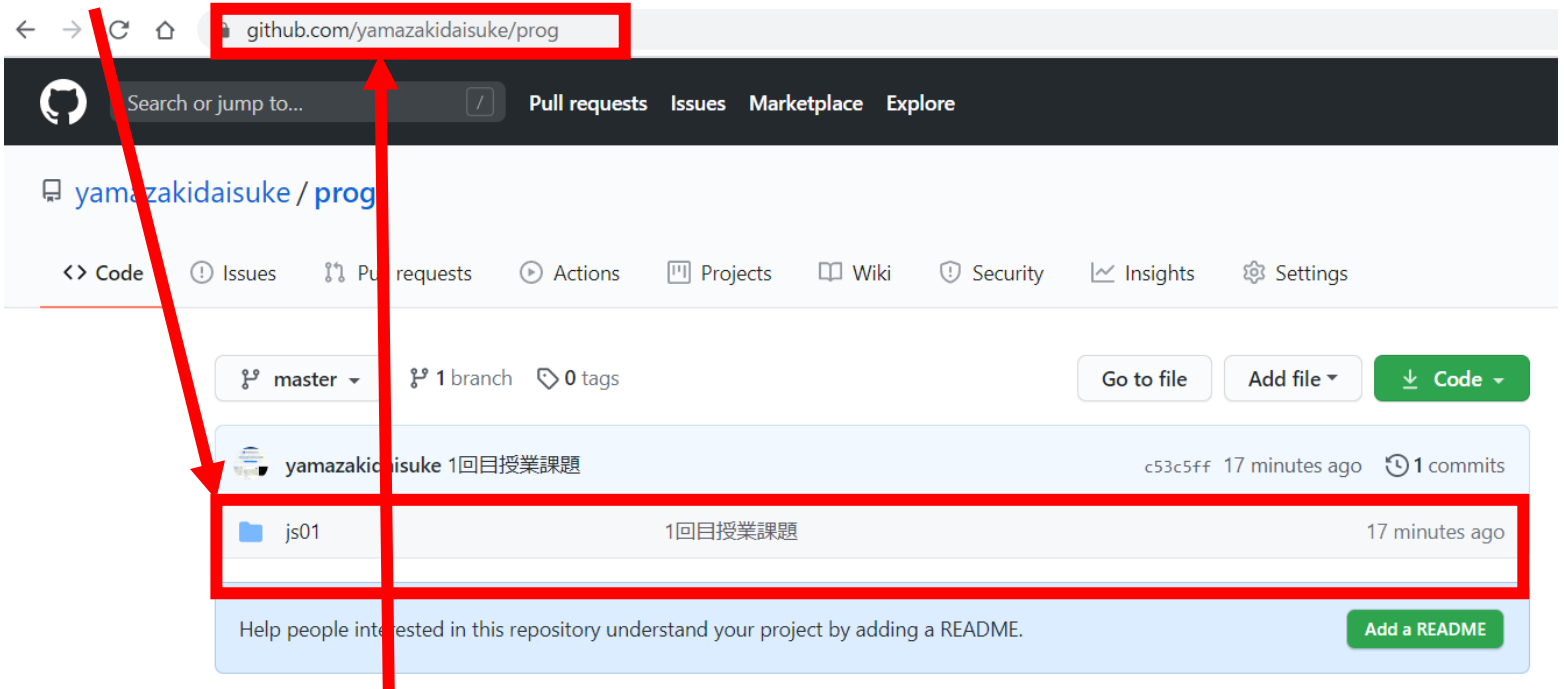
授業制作物：提出の方法4

1. デスクトップに作成した「01」フォルダをドラッグ＆ドロップでアップロード



授業制作物：提出の方法5

1. 「01」と表示されていれば提出完了



2. このページのURLを以下Googleフォームに登録

https://docs.google.com/forms/d/e/1FAIpQLSeARKcS-WTUvsrz1Gs-9XkXqiUe2DhDpoeByzWFUMgY9mHiAg/viewform

締め切り

翌週授業 授業開始時まで

今日作成する部分

Fileアップロードからデータベース登録の流れ

1.画面作成
2.画像選択（カメラ起動）ボタン
3.送信ボタン
4.ファイル選択ボタンのデザイン変更
5.4の対応に合わせたスクリプト追加
6.FileAPIを利用したサイズ制限

post送信

1. **POST/Fileデータ受信**
2. **画像をuploadフォルダへ保存**
3. 画像ファイル名を変更（Uniq値）
4. 画像名・緯度・経度・登録日時データベースへ保存

ファイル選択を学ぶ

～Camera／Audio選択～

画像写真選択とは

資料のHTML/css/javascript文字はコピーしないように。
PDF変換で変な見えない文字が入ってるのでエラーの元です。

◇ 写真選択：form要素と属性

```
<form method=“③” action=“送信先" enctype=”②">
  <input type=“①" accept="image/*”  name="upfile”>
  <input type="submit” value="Fileアップロード”>
 </form>
```

① FILE選択できるようにする

<input type=“file” ……>

※他の例<input type=“text” ……>

② File送信時はenctype属性を指定

enctype="multipart/form-data”

③ POST送信（Action=“送信先”）

Camera／写真選択とは

資料のHTML/css/javascript文字はコピーしないように。
PDF変換で変な見えない文字が入ってるのでエラーの元です。

◇ Camera/写真選択：accept属性

```
<form method="post" action="送信先" enctype="multipart/form-data">
  <input type="file" accept="image/*" capture="camera" name="upfile">
  <input type="submit"  id="upload_btn" value="Fileアップロード”>
 </form>
```

② カメラ起動＆画像選択可能

input accept="image/*" capture="camera"

※ 他の記述方法例1）accept="image/jpeg, image/gif, image/png"

※ 他の記述方法例2）accept="audio/*"

※ 他の記述方法例3）accept="video/*"

※ 他の記述方法例4）accept="text/comma-separated-values"

File API

FileAPIとは

File API

◇FileAPI

HTML5で定義されたファイル操作に関するAPIです。File APIのFileオブジェクトを利用することで、ローカル（PC内）にある『ファイル情報』や『データ本体』の読み込みができます（テキストファイル、画像ファイルなど）。
JavaScriptで直接ローカルにあるファイルのデータを処理して結果をユーザーに提示できるので、ブラウザー上のJavaScriptのみで完結するWebアプリの可能性が広がります。

◇現在File APIに対応しているブラウザーのバージョンです。

ブラウザー覧	対応バージョン
Internet Explorer	10以降
Firefox	3.5以降
Google Chrome	6以降
Safari	5以降
Opera	10以降

◇File情報を取得するプロパティ（ file_readinfo.html ）

プロパティ	説明
name	名前
type	コンテンツタイプ
size	サイズ（バイト単位）
lastModifiedDate	最終更新日時

POINT
大事！

File API

File情報を取得・表示

サンプル
file/file_readinfo.html

FileAPI:情報取得

1. FileAPIが使用できるか確認させる(手で打ちましょう！)

```
$('#file').on('change', function() {
    if (window.File &&
        window.FileReader &&
        window.FileList &&
        window.Blob
    ) {
            alert("FileAPIが使用可能！");
   }
});
```

サンプル
file/file_readinfo.html

FileAPI:情報取得

1. FileAPIが使用できるか確認させる

```
$('#file').on('change', function() {
    if (window.File &&
        window.FileReader &&
        window.FileList &&
        window.Blob
    ) {
            alert("FileAPIが使用可能！");
            const input = $('#file').get(0).files[0];
    }
});
```

FileAPI:情報取得

サンプル
file/file_readinfo.html

1. FileAPIが使用できるか確認させる

```
$('#file').on('change', function() {
    if (window.File &&
        window.FileReader &&
        window.FileList &&
        window.Blob
    ) {
        const input = $('#file').get(0).files[0];
        $('#name').html(input.name);
        $('#type').html(input.type);
        $('#size').html(input.size / 1024);
        $('#lmd').html(input.lastModifiedDate);
    }
});
```

ここの4行追記

File API

File画像を読み込み取得・ライブラリ紹介

サンプル
file/file_readimage.html

FileAPI:画像読み込み

1. FileAPIが使用できるか確認させる(手で打ちましょう！)

```
$('#file').on('change', function() {
    if (window.File &&  window.FileReader &&
        window.FileList && window.Blob
    ) {
        const input = $('#file').get(0).files[0];          //2
        const reader = new FileReader();                   //3
        $(reader).on('error',function () {                 //6
            alert("読み取り時にエラーが発生しました。");
        });
        $(reader).on('load', function() {                  //5
            $('#result').attr('src', reader.result);
        }
        reader.readAsDataURL(input);                       //4
    }
});
```

ここを追記

+α 画像加工ライブラリ1

サンプル
file/file_readimage.html

CamanJS
http://camanjs.com/

```
<script src="http://cdnjs.cloudflare.com/ajax/libs/camanjs/4.0.0/caman.full.min.js">
</script>


 Caman('#result', function () {
            this.brightness(10);
            this.contrast(30);
            this.sepia(60);
            this.saturation(-30);
            this.render();
            this.noise(50);
});
```

+α 画像加工ライブラリ2

オススメ

Pixastic
http://dph.am/pixastic-docs/

```
<script src="pixastic.custom.js"></script>
<script>
var can = document.getElementById("demoimage");
var context = can.getContext("2d");
//Image
var img = new Image();
img.src = "base.jpg";
img.onload = function(){
    context.drawImage(img,0,0);
};
function demo() {
      Pixastic.process(can, "blur");
}
```

File API

Fileテキストを読み込み・表示

サンプル
file/file_readview.html

FileAPI:テキスト読み込み

1. FileAPIが使用できるか確認させる(手で打ちましょう！)

```
$('#file').on('change', function() {
    if (window.File &&  window.FileReader &&
        window.FileList && window.Blob
    ) {
        const input = $('#file').get(0).files[0];          //2
        const reader = new FileReader();                   //3
        $(reader).on('load', function() {                  //5
            $('#result').val(reader.result);
        }
        reader.readAsText(input, 'UTF-8');                 //4
    }
});
```

ここを追記

+α グラフ化

サンプル
file/file_graph.html

Chart.js
http://www.chartjs.org/

授業内で解説

File API

サンプル
file/file_readview.html

サンプル
file/file_readimage.html

◇ ファイルの読み取り

FileReaderオブジェクト	FileReaderオブジェクト内容
FileReader.readAsText (Blob|File, opt_encoding)	result プロパティにはファイル/ブロブ データが**「テキスト文字列」**として格納されます。デフォルトでは、この文字列は「UTF-8」としてデコードされます。オプションのエンコード パラメータを使用すると、他の形式を指定できます。
FileReader.readAsDataURL (Blob|File)	result プロパティにはデータ URL としてエンコードされたファイル/ブロブ データが格納されます。**（画像/音声）**
FileReader.readAsArrayBuffer (Blob|File)	result プロパティには、ファイル/ブロブ データが ArrayBuffer オブジェクトとして格納されます。**（テキスト文字列でない）**

◇ その他のメソッドとプロパティ

プロパティ名	内容
state	読み込み処理の状態を取得する（読み取り専用）
result	読み込み成功後に、中身のデータを取得する。（読み取り専用） ※resultは readAsTextなど上記FileReaderオブジェクト を実行した後、onloadイベントのタイミングで取得できます。
abort	読み込みを破棄する

File API

サンプル
file/file_graph.html

◇ FileReaderのイベント

プロパティ名	内容
onloadstart	読み込み開始
onprogress	読み込み中
onabort	読み込みを破棄する
onerror	エラーが発生した
onload	読み込み成功して完了
onloadend	読み込み完了（エラーを含む）

◇ ブラウザ対応チェック処理

```
if (window.File && window.FileReader && window.FileList && window.Blob) {
        // サポートしてるので、処理をここに記述
} else {
      alert('ブラウザに対応してません');
}
```

◇ サンプルファイル一覧

サンプルファイル名	**サンプル操作で使用するファイル**
file_readinfo.html	data/responsive.jpg
file_readview.html	data/file_readview.txt
file_readimage.html	data/responsive.jpg
file_graph.html	data/chart_data.txt

FileSizeチェック

camera.html

FileAPI：写真アップロード制限（１）

camera.html

◇ファイルアップロード画像の制限をつける
　FileAPIを利用することで可能になる！

◇例１：FileSize情報取得（jQuery利用した場合）

```
const file = $(ターゲット).get(0).files[0];
console.log(file.size);       //ファイルサイズ取得
console.log(file.name);       //ファイル名取得
console.log(file.type);       //ファイル・タイプ取得[image/jpeg]
```

◇例２：FileSize情報取得（javascriptの場合）

```
const  = document.querySelector(ターゲット).files[0];
console.log(file.size);       //ファイルサイズ取得
console.log(file.name);       //ファイル名取得
console.log(file.type);       //ファイル・タイプ取得[image/jpeg]
```

FileAPI：写真アップロード制限（２）

camera.html

◇ファイルサイズを取得したら分岐処理！

```
if ( file.size/1024 > 150 ) {  //150kbyte(0.15M)
    alert("OVER");
    return false;
} else {
    console.log("OK");
}
```

解説：file.size/1024

ファイルサイズはbyteなので、１kであればfile.size=1024となります。「1024」で割ることで、解りやすく「 Kbyte 」キロバイトで条件を考えることができます。
条件に満たない場合には「 return false; 」で処理終了とする。
※File Uploadボタンは表示されません。

FileAPI：写真アップロード制限（3）

camera.html

◇ファイルTypeを取得したら分岐処理！

```
if ( file.type != "image/jpeg" ) { //jpgでなければalert
    alert("OVER");
    return false;                          //処理をここでSTOP
}
```

解説：file.type

JPEG　→　image/jpeg
GIF　→　image/gif
PNG　→　image/png
上記３つのようなfile属性を取得できます。

課題

file_chek.html

課題

４つの課題（条件と表示）をクリヤーしてください。

#select_btn

1. 画像選択＆ jpeg画像のみに制限する
※png , gif はNGとする。

2. alertではなく「h2要素」に
「jpeg以外はアップロードできません」と表示

3. データ量は「1.5M」の制限とする。

4. alertではなく「h1要素」 に
「FileDataOver!!」と赤文字で表示。

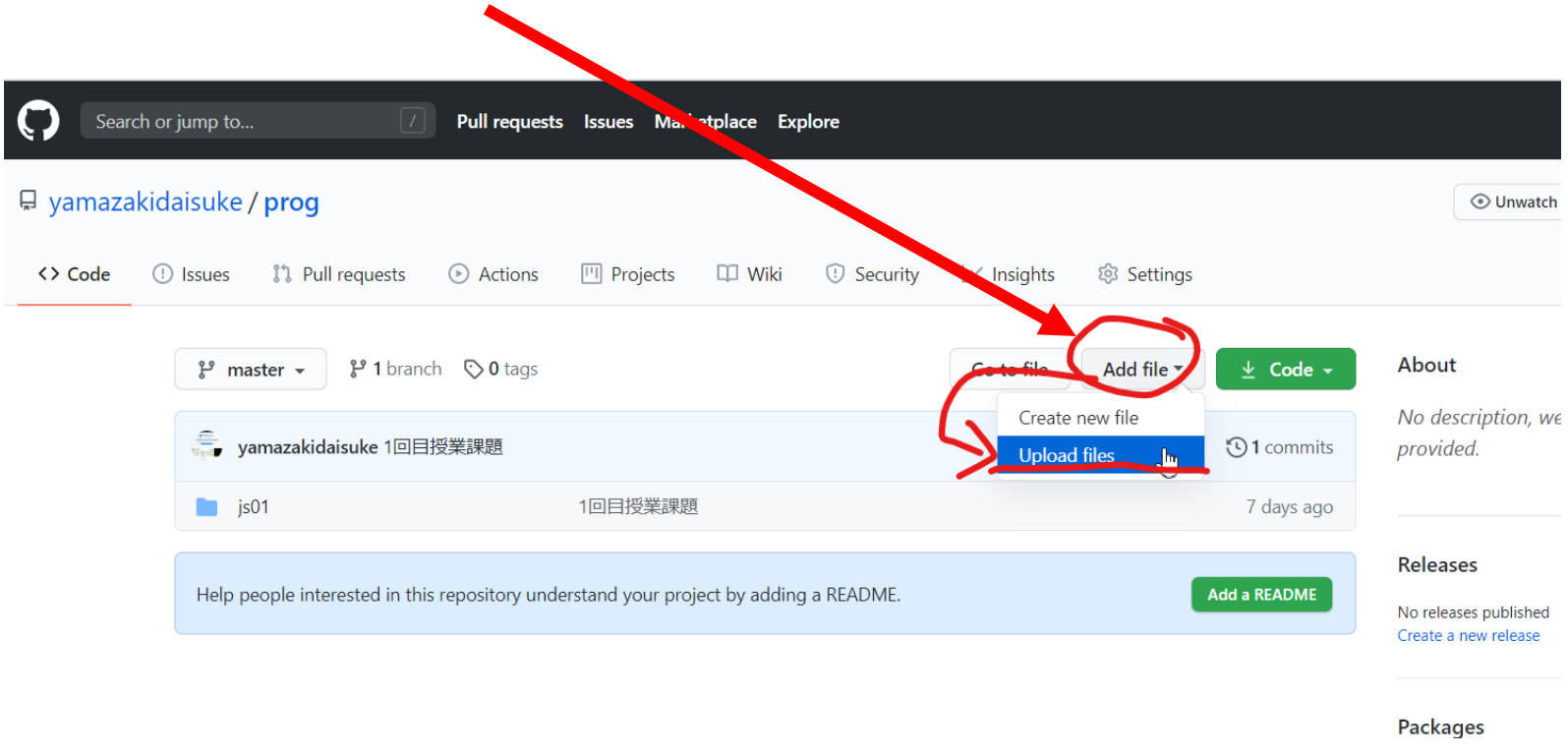
授業制作物：提出の方法

★発展自由：評価対象です
なにか面白いこと・新しいことにチャレンジしてきたら言ってください！！
発表してもらいますし、評価も高くなります！！

締め切り
翌週授業（金曜日）授業開始時まで

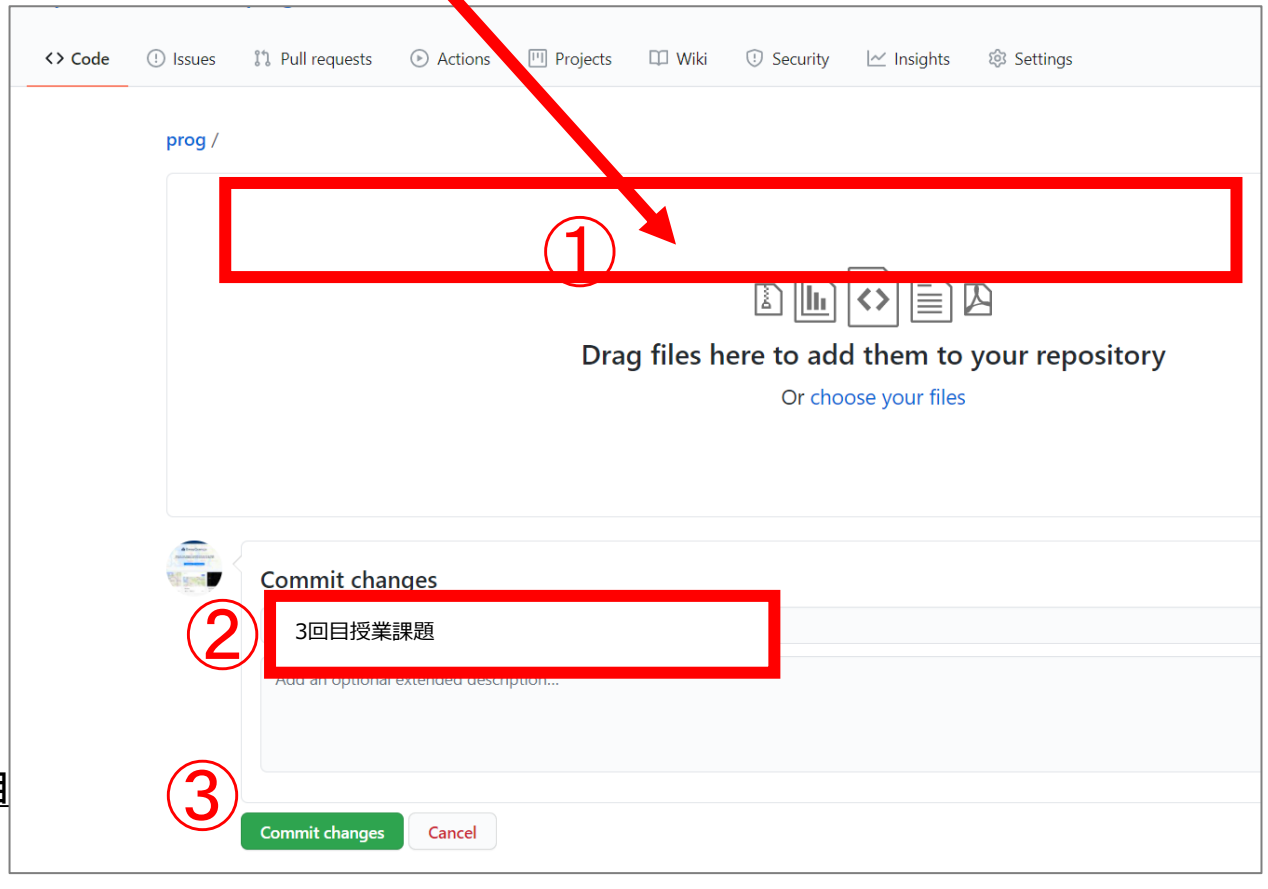
授業制作物：提出の方法1

1. 前回作ったprogリポジトリのURL画面を開きます。
2. 「Add file」→「Upload files」を選択



授業制作物：提出の方法2

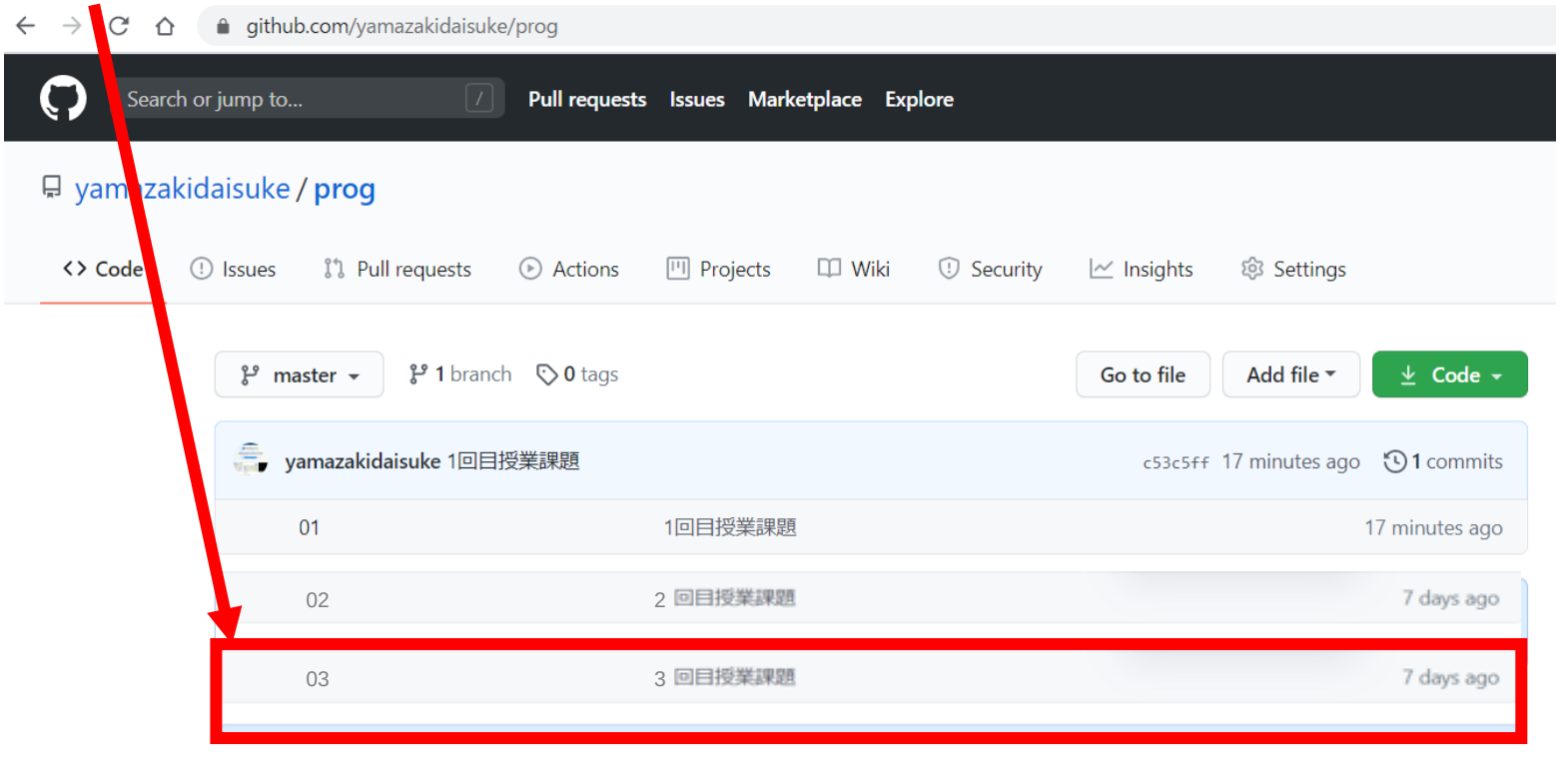
1. デスクトップに作成した「03」フォルダをドラッグ＆ドロップでアップロード



★発展自

授業制作物：提出の方法3

1. 「 03 」と表示されていればOK



提出完了！！