
スマホで撮った動画シェア
もうこんなに簡単なんだ。
どうやるの？
Google+
フォローする
フォロー
617
Sponsors
結城浩さんからのアルゴリズム問題
解答すると、直接フィードバックがもらえます
[広告募集中]
150 × 150
[広告募集中]
150 × 150
[広告募集中]
150 × 150
» 広告のお問い合わせは
広告
中古再生サーバ
ネットワーク機器
Ajaxを利用する上で基本的な仕組みについて解説します。まずはAjaxの基本的要素であるHTTP通信を使ってHTMLファイルとサーバ間での通信を行う方法を確認し、最後にクライアント側のHTMLファイルとサーバ間で通信を行う際にクライアント側のHTMLファイルとの各種連携方法を見ていきます。
なおAjaxを利用する上でJavaScriptを使います。JavaScriptについては『JavaScript入門』を参照して下さい。
Ajaxを用いたHTTP通信
・XMLHttpRequestオブジェクト
・XMLHttpRequestオブジェクトのメソッドとプロパティ
・同期通信によるファイルの取得
・非同期通信によるファイルの取得
・InternetExplorer(IE)での非同期通信

図 1


Ajaxの基本
TOP
1章
1-1 HTTP通信の流れ
1-2 ブラウザの互換性
1-3 タイムアウト処理
1-4 エラー表示
1-5 DOM
1-6 イベント
1-7 マウスイベント
1-8 キーイベント
2章
2-1 Ajax
2-2 ラジオボタン
2-3 テキストボックスの入力補助
readyStateプロパティの値
HTTPレスポンスコード
イベント一覧
マウスイベント一覧
キーボードイベント一覧
keyCode一覧
このサイトはAjaxの基本を勉強するサイトです。
このサイトを見て勉強する場合の作成環境をいくつか指定させてもらいます。
• ブラウザ：Firefox
• ソフト：Jedit
• 文字コード(エンコーディング)：UTF-8
• 改行タイプ：Unix(LF)
• テキストタイプ：プレインテキスト
• xamppに作成したファイルを置き、htmlファイルをブラウザで実行する
以上の環境でないと説明した通りに表示されない場合があります。
プログラムを作成してもうまく表示されない、エラーがでるなどした場合は、Firefoxのエラーコンソール機能を使いチェックしてください。
やり方がわからない場合は下記のリンクをクリックしてください。
HTMLの文法チェックサイトも同時に載せておきます。
エラーコンソールの使い方 | HTML文法チェック
なお、各項目の解説と問題へのリンクはページ上部と下部にあります。
※ このサイトはJavaScriptの知識をある程度もっていることを前提に進めていきます。
Ajaxとは
Yahoo!の検索入力補助機能や地図などページの一部のみを書き換える非同期通信技術のことです。もっと詳しく知りたい場合は下記のサイト参考してください。
Ajax参考サイト
Ajax入門|Ajaxの基礎と応用の学習|Study Net | Ajax初心者入門講座 | Wikipedia-Ajax

図 2

図 3

本入門は以下の環境を前提としています。

※PHPを使用するため、必ずローカルサーバーを起動しておいてください！

ブラウザ	FireFox・Safari
文字コード	UTF-8
サーバーソフト	XAMPP・AN HTTPD
サーバーサイドプログラム	PHP

ブラウザ

Ajaxはブラウザによって記述が違います。 **本入門のAjaxプログラムはInternetExplorer（以下IE）には対応していません。** IEに対応させる方法はL.4 クロスブラウザ対応を参照してください。

サーバーソフト

XAMPP・AN HTTPDは一例です。本入門ではPHPを使用するので、それに対応した サーバーソフトならば、なんでも構いません。

サーバーサイドプログラム

文字通り、サーバー側で使用するプログラミング言語です。 Ajaxによってサーバーに送信されたリクエストに対する操作は、この言語で行ないます。 使用する言語に決まりはありませんが、本入門では初心者でも扱い易いPHPを採用しています。

図 4

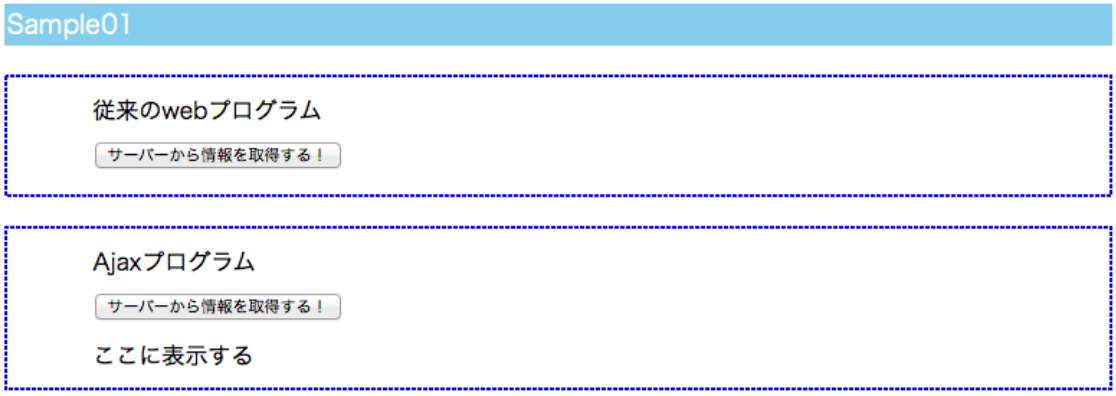
Sample01
従来のwebプログラム
サーバーから情報を取得する！
Ajaxプログラム
サーバーから情報を取得する！
ここに表示する

図 5

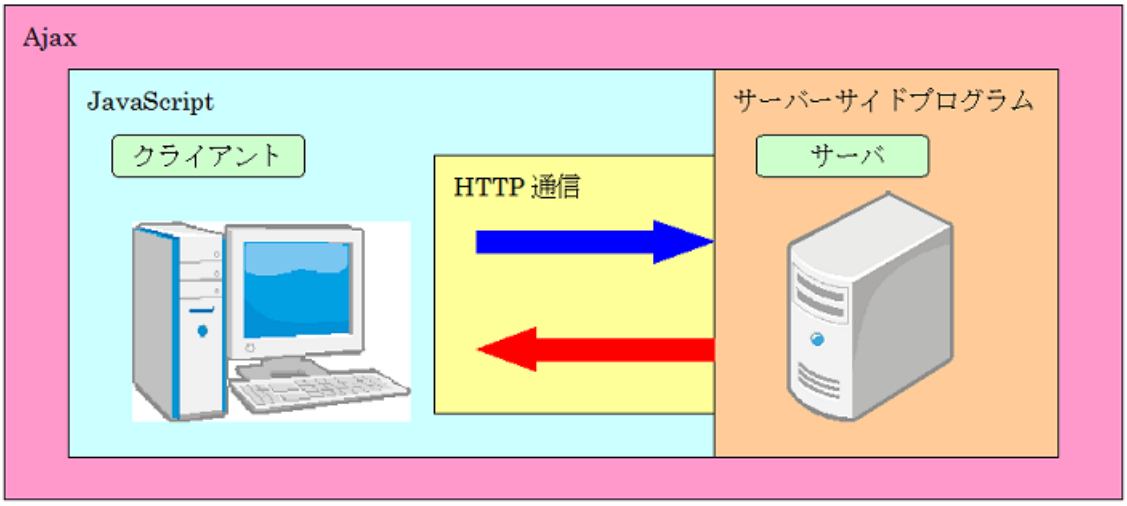
Ajax
JavaScript
クライアント
HTTP 通信
サーバーサイドプログラム
サーバ

図 6

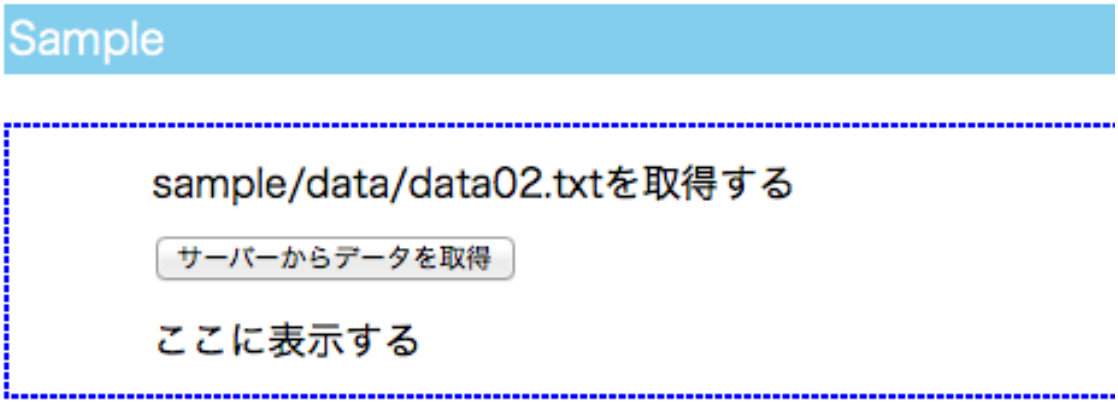

図 7

JavaScript解説

JavaScriptの関数

関数とは一連の処理をひとまとめにして何度も呼び出せるようにしたものです。 例えば以下のように記述します。

```
function test(){
        alert("テスト");
        }
test();
```

関数の作成にはfunctionと記述し、半角スペースあけて関数名()と記述します。 そして {} の中に処理を記述します。関数を呼び出すには上記の例だと、「test();」 とします。実行するとtest関数の中身はalert("テスト")なので「テスト」と書かれたダイアログが出現します。

図 8

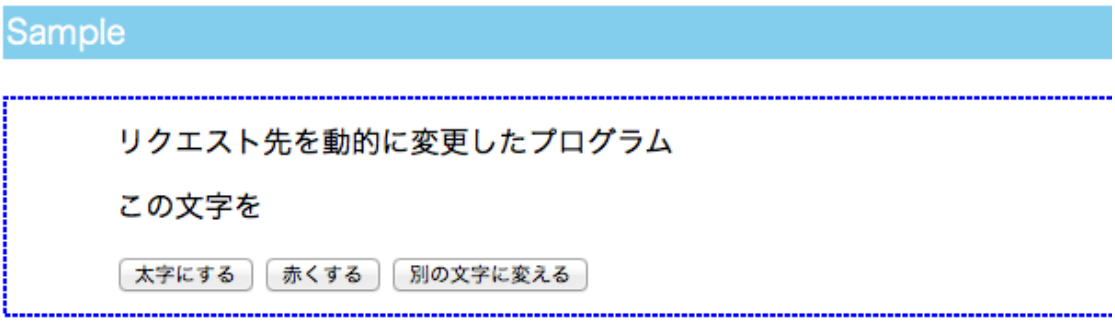
Sample
リクエスト先を動的に変更したプログラム
この文字を
太字にする
赤くする
別の文字に変える

図 9

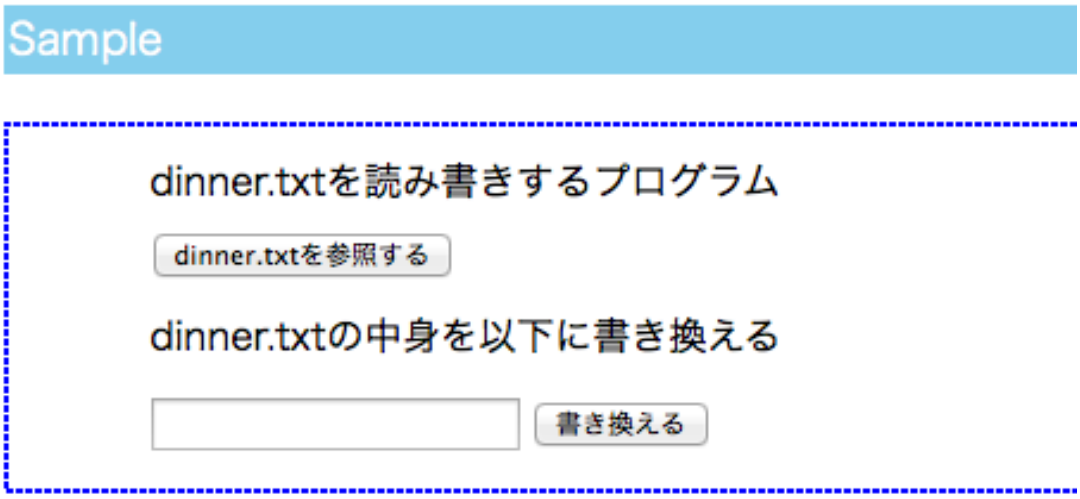
Sample
dinner.txtを読み書きするプログラム
dinner.txtを参照する
dinner.txtの中身を以下に書き換える
書き換える

図 10

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; UTF-8">
<title>課題</title>
</head>
<body>
<center>

<!--            ここから下を編集                    -->

<p>Ajaxプログラム</p>
<input type="button" value="データを取得">
<p id="chtext">ここに表示する</p>

</body>
<script type="text/javascript">
```

実行

解答を見る

```
<p>Ajaxプログラム</p>
→  <input type="button" value="データを取得" onclick="SendRequest()">
<p id="chtext">ここに表示する</p>

</body>
<script type="text/javascript">

var request;

function GetResponse(){
→      document.getElementById("chtext").innerHTML = request.responseText;
       }

function SendRequest(){
→      request = new XMLHttpRequest();
→      request.onload = GetResponse;
→      request.open("GET","sample/data/data03.txt");
→      request.send(null);
       }
```

閉じる

図 11

```
30
31 }
32
33 </script>
```

Check Point

任意の変数にXMLHttpRequestオブジェクトを生成する。

```
26 request = new XMLHttpRequest();
```

onloadプロパティにレスポンス受信時に呼び出される関数を設定する。(ここでは16行目に作成したGetResponse関数)

```
27 xmlhttp.onload = GetResponse;
```

リクエストの送信方法と送信先を設定する。第一引数：送信方法　第二引数：送信先URL

```
28 xmlhttp.open("GET","sample/data/data02.txt");
```

図 12(改善前)

```
31 }
32
33 </script>
```

sample/data/data02.txt

sample/data/data02.txtを取得しました！

Check Point

JavaScriptは「<script type="text/javascript"></script>」の中に 記述します。

①XMLHttpRequestオブジェクトを生成する

```
26  request = new XMLHttpRequest();
```

オブジェクトの機能を活用するには、「**new** オブジェクト名()」という形で生成する必要があります。 今回は16行目で定義した変数「request」にXMLHttpRequestオブジェクトを生成しています。これによりrequestを通じて XMLHttpRequestが持つメンバを利用することができます。

②レスポンス受信時の処理を設定する

```
18  function GetResponse(){
27  request.onload = GetResponse;
```

「onload」にはレスポンス受信時の処理を設定します。今回はGetResponseを指定しています。これは18行目で作成した関数です。この関数の処理内容は

図 13(改善後)