

## 4.2 FORM(フォーム)の作成

### クライアント側入力画面

= ユーザと対話(情報入力や選択)を行うページの作成 (HTML, CSS などで作成)

↓

サーバ送信

↓

### サーバ側での処理

= メール, 掲示板, SNS, データベース...

= スクリプトなどプログラミング要 (PHP, CGI, ... が動作)



実習) 教科書サンプルデータの準備

→ 復習: 圧縮ファイル

### form 要素

<form > ~ </form>

- ・サーバに情報を送信するためのフォーム領域を作成する
- ・基本的にはこの要素内に form の部品を配置する  
(HTML5 では id 属性を使用して form の外にも配置可能)
- ・入力値は「送信」ボタンでサーバに送られる



#### ・属性

**method**

- GET or POST

GET: データを URL 形式で受け渡す

POST: データを本文 (contents) で受け渡す ← 制限なし

**action**

- URL, 入力を送信する先のアドレス

確認) action などの属性値を指定しなかった場合は? (→ form のページに戻る)

### フォームの部品要素

関連・復習) Windows 基礎知識, コンピュータ概論, 画面設計(インターフェース設計)

テキストボックス

→ text, password, textarea

ラジオボタン

→ radio

チェックボックス

→ checkbox

リストボックス(プルダウンメニュー)

→ select

ボタン

→ submit, button

プログレスバー

→ progress

↓

とりあえず input, select, submit 要素を中心に

- input** 要素
- フォームの部品(多種多様), 空要素
  - ・ 属性
    - type**
      - 部品のタイプを指定する
        - text**: 文字入力
        - password**: 文字入力(ただし入力内容はマスクされる)
        - radio**: ラジオボタン
        - checkbox**: チェックボックス
        - email**: 電子メールアドレスを入力
        - file**: 送信ファイルを入力
        - submit**: サーバに送信
        - reset**: 入力リセット(クリア)
        - button**: 汎用ボタン
        - hidden**: 表示されない(直接入力できない)
        - :
        - :
    - name**
      - 部品の名前
        - サーバ側には 「 name=入力値 」 の形式で送信される
        - プログラミングの「代入文」的?
        - これを受け取ってサーバ側の処理を行う
    - value**
      - 値の初期値などを指定する
  - その他
    - required**
      - 送信しなければならない項目に設定する
    - disabled**
      - 対話(入力・選択)できない状態にする、送信もされない
    - autofocus**
      - フォーカスの設定, 1つの要素だけ指定可
    - checked**
      - radio, checkbox のデフォルト値に設定する
- select** 要素
- フォームの部品(プルダウンメニュー)
  - <select>~</select> 要素内に <option>~</option> 要素で選択肢を記述する
  - select の属性は input の name, その他に準じる
- option** 要素
- 選択肢の記述
  - 属性
    - value**
      - 選択された場合の select(name) の値
    - selected**
      - select のデフォルト値として設定する

#### 実習) フォームを作ってみよう

- ・ それぞれの部品について確認しよう
- ・ サーバ側の処理に何が必要なのか?調べてみましょう