



Woodland
Academy

TeamProject補足

- reset.css
- Sessoinとは



Shape Your Future

目次

1 reset.cssとは

2 Sessionとは

reset.cssとは

- ブラウザによって異なるデフォルトのCSSを打ち消してブラウザ間の表示を揃えるためのCSSファイルのこと。

✓リセットCSSを使うメリット

- ブラウザごとの差異をなくすることができる。
- ブラウザ固有のバグを修正することができる。
- 自分でブラウザごとのCSSを書かなくて済む。

reset.cssの使い方

- リセットCSSを使う場合は、自作のCSSの前に必ず読み込む

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>sample</title>
  <link rel="stylesheet" href="./css/reset.css">
  <link rel="stylesheet" href="./css/style.css">
</head>
```

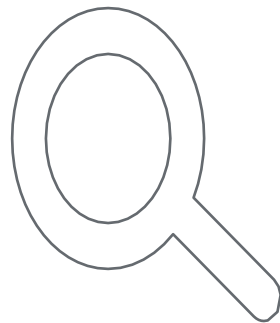
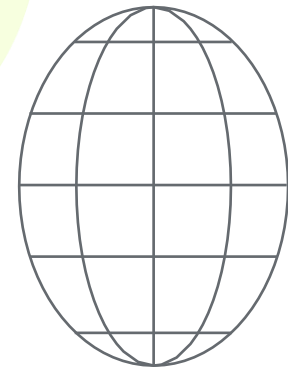
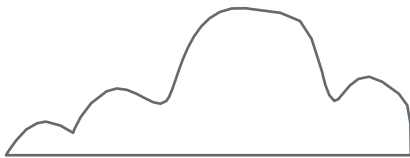
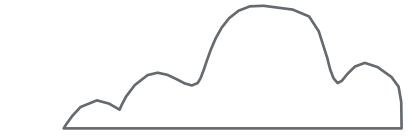
基本的にリセットCSSはファイルをコピーして、HTMLファイルやCSSファイルと同様に自分のサーバー上に配置することが多いが、直接外部サーバーのファイルをHTML上から読み込むことも可能。

おすすめのreset.css

- 以下のサイトからチーム内で話し合って好きなreset.cssを使用してください。
- デフォルトスタイルがないreset.css
 - [The New CSS Reset](#)
 - [destyle.css](#)
- デフォルトスタイルがあるreset.css
 - [Normalize.css](#)
 - [A modern CSS reset](#)



Q&A



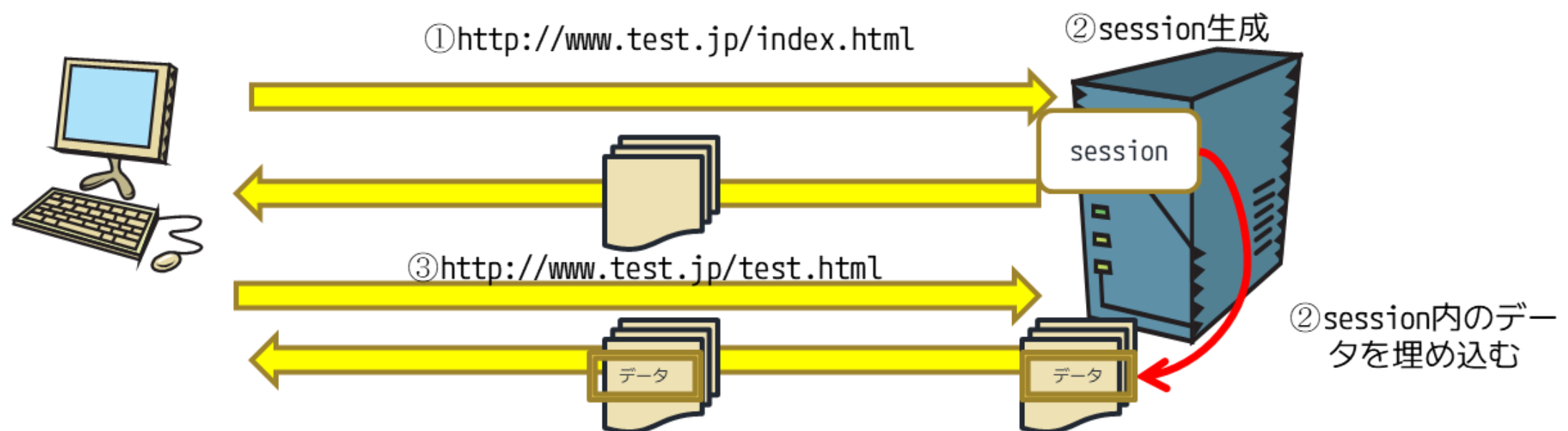
目次

1 reset.cssとは

2 Sessionとは

Sessionとは

- セッション：ユーザ情報を，一時的にサーバ側で保持される情報のこと．例えばショッピングカート。
- ページをまたいでも，データが保持される．**どのページでも共通のデータを使いたい時**に用いる機能。
- (名前, 値)という形式で保管される。



Sessionの基本構文

- 【セッションオブジェクトの宣言】

@Autowired

HttpSession **session**;

- 【セッションに値を格納】

session.setAttribute("属性名 (key) ", "値 (value) ")

- 【セッションに格納されている値の取得】

session.getAttribute("属性名 (key) ")

- 【セッションの無効化】

session.invalidate()

Sessionの使い方①

- ・【セッションオブジェクトの宣言】

@Autowired

HttpSession session;

```
@Controller
@RequestMapping("/admin/")
public class AdminLoginController {
    @Autowired
    AdminService adminService;
    /**
     * ログイン済みユーザ情報を格納するための
     * セッションオブジェクト
     */
    @Autowired
    HttpSession session;
```

Sessionの使い方②

- 【セッションに値を格納】

`session.setAttribute("属性名 (key) ", "値 (value) ")`

```
@PostMapping("/login")
public String adminAuth(@RequestParam String adminEmail, @RequestParam String password, Model model) {
    //adminServiceクラスのfindByAdminNameAndPasswordメソッドを使用して、該当するユーザー情報を取得する。
    AdminEntity adminEntity = adminService.findByAdminEmailAndPassword(adminEmail, password);
    //adminEntity == null
    if(adminEntity == null) {
        //admin_login.htmlが表示される。
        return "/admin/admin_login.html";
    }else {
        //adminEntityの内容をsessionに保存する
        session.setAttribute("admin", adminEntity);
    }
}
```

Entityの内容をセッションに
セットしている

Sessionの使い方③

- 【セッションに格納されている値の取得】
session.getAttribute("属性名 (key) ")

```
//adminEntityの内容をsessionに保存する
session.setAttribute("admin",adminEntity);

//adminServiceクラスのselectFindAll()を使用して、一覧を取得する。
List<AdminEntity>adminList = adminService.selectAdminAll();
//adminListをキーにしてadminListをitem_list.htmlに渡す。
model.addAttribute("adminList",adminList);

//セッションから管理者の情報を取得する。
AdminEntity auth = (AdminEntity) session.getAttribute("admin");
//管理者情報(AdminEntity)から管理者名を取得する。
String loginAdminName = auth.getAdminName();

model.addAttribute("loginAdminName",loginAdminName);
return "/admin/admin_all_view.html";
```

↑ セッションに格納している値の取得をしている。

Sessionの使い方④

- 【セッションの無効化】
session.invalidate()

```
//ログアウト処理
@GetMapping("/logout")
public String adminLogout() {
    //セッションの情報を削除する。
    session.invalidate();
    //admin_login.htmlが表示される。
    return "/admin/admin_login.html";
}
```

Sessionの無効化
ログアウト処理などに使用

まとめ

Sum Up



1.Reset.cssの使い方と概念。

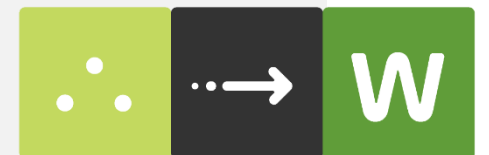
2.セッションとは

① セッションの概念

②セッションの書き方

Thank you!

From Seeds to Woodland — Shape Your Future.



Shape Your Future