

# PHP SESSIONS

รวบรวมโดย อ. ฐาปนพงษ์ รักกาญจน์

# Information Flow

- จากที่เรียนมา การส่งข้อมูลระหว่าง **Web Page** จะต้องอาศัย **Form** เป็นตัวกลางในการส่งข้อมูล ซึ่งถ้าหากข้อมูลต้องการนำมาใช้ในทุก **Web Page** แล้วนั้น เราจะต้องสร้าง **Form** ทุกครั้ง อีกทั้งไม่สามารถใช้ `<a></a>` ในการลิงค์ไปยังหน้าอื่นได้ (เนื่องจากข้อมูลจะไม่ถูกส่งอย่างต่อเนื่อง)
- สำหรับ **Web Application** เราสามารถจัดการกับปัญหานี้ได้โดยการให้ **Web Browser** จำข้อมูลเหล่านั้น ซึ่งการจำสามารถทำได้สองแบบคือ **Cookie (Java Script)** และ **Sessions (PHP)**

# Memorization

- การจดจำข้อมูลในรูปแบบของตัวแปร เราจะใช้ระบบ **Sessions** ที่ **PHP** สร้างขึ้น
- ผู้เข้าชมเว็บของเราจะได้รับ **Session ID** หรือมักจะเรียกว่า **Unique ID (UID)** ซึ่งจะถูกรับที่ลงใน **Cookie** หรือส่งต่อผ่าน **URL**
- **PHP Sessions** อนุญาตให้เราบันทึกข้อมูลลงใน **Superglobal Array** ที่ชื่อว่า **\$\_SESSION**

# Sessions Preprocessing

- ก่อนที่เราจะส่งให้ **PHP Sessions** จดจำข้อมูลของเรา เราจะต้องทำการเรียกใช้ **PHP Sessions** ก่อนด้วยคำสั่ง
  - `session_start();`
- คำสั่งข้างต้นเป็นการส่งให้เว็บของเราเริ่มการทำงานของ **PHP Sessions** ซึ่งจะต้องประกาศไว้ในส่วนแรกสุดของ **ทุกไฟล์ PHP** ที่ต้องการใช้หรือบันทึกข้อมูลที่จดจำไว้

# Information Storing

- หลักจากเราเรียกใช้งาน **PHP Sessions** เราสามารถบันทึกข้อมูลลงใน **Superglobal Array \$\_Session** ดังตัวอย่างด้านล่าง

```
<?php
```

```
session_start(); // เรียกใช้ PHP Sessions
```

```
$_SESSION['username'] = "user001";
```

```
$_SESSION['age'] = 20;
```

```
?>
```

# Information Retrieval

- หลักจากที่เราบันทึกข้อมูลลงใน **Superglobal Array** **\$\_SESSION** แล้วนั้น เราสามารถดึงข้อมูลมาใช้ในโดยไม่ต้องส่งข้อมูลผ่าน **Form** อีก

File1.php

```
<?php  
session_start();
```

```
$_SESSION['username'] = "James";  
$_SESSION['age'] = 25
```

```
// บันทึกข้อมูลใน Sessions
```

```
?>
```

File2.php

```
<?php  
session_start();
```

```
echo $_SESSION['username'];  
echo $_SESSION['age'];
```

```
// เอาข้อมูลที่บันทึกมาใช้
```

```
?>
```

# Existence Validation

- ❑ ข้อควรระวังในการเรียกใช้ Superglobal Array `$_SESSION` คือ การเรียกใช้ข้อมูลที่ยังไม่เคยมีการบันทึกมาก่อน PHP จะส่ง Error **Undefined Index** กลับมา ดังนั้นก่อนการเรียกใช้ข้อมูลที่เราไม่แน่ใจว่ามีการบันทึกมาก่อน เราควรตรวจสอบก่อนว่า ข้อมูลนี้มีการบันทึกมาแล้วหรือไม่ด้วยฟังก์ชัน `isset`

```
if(isset($_SESSION['username']))  
  
    echo $_SESSION['username'];
```

# Terminate Sessions

- ถ้าหากเราต้องการบันทึกข้อมูลนี้เลย หรือต้องการทำลายข้อมูลที่บันทึกไว้ เราสามารถทำได้ด้วยคำสั่ง
  - `session_unset();` // ลบตัวแปรทั้งหมด
  - `session_destroy();` // จบการทำงานของ Sessions



# Login System

- ระบบ Login เป็นระบบที่เว็บไซต์จะทำการจดจำการลงชื่อเข้าใช้ หรือที่เราเรียกว่าการ login ซึ่งเว็บไซต์จะต้องจำข้อมูลที่สำคัญ สำหรับการยืนยันตัวตนของผู้ใช้เช่น Username, E-mail ข้อมูลที่ไม่ควรจะจดจำคือรหัสผ่าน ข้อมูลส่วนตัว
- ระบบ Login จะประกอบไปด้วยสองส่วนหลัก ๆ คือ
  - ▣ Login.php ทำหน้าที่ตรวจสอบว่า Username และ Password ได้ผ่านการลงทะเบียนหรือไม่ จากนั้นทำการบันทึก Username ลงใน Sessions
  - ▣ Logout.php ทำหน้าที่ลบข้อมูลการ Login ออกจากระบบ

# Login System (2)

## Homepage.php

ถ้ายังไม่ Login ให้แสดงฟอร์มเพื่อ Login

```
<form method=POST action=login.php>
```

...

```
</form>
```

ถ้า Login แล้ว ให้แสดงข้อความต้อนรับ และ  
ลิงค์เพื่อ Logout

```
echo "Hello, $_SESSION['username']";
```

```
echo "<a href=logout.php> logout </a>";
```

## Login.php

- ❖ ตรวจสอบว่ามี การ Login แล้วหรือไม่
- ❖ ตรวจสอบว่ากรอก Username Password ถูกต้องหรือไม่ (ตรวจว่า Username นี้ได้ทำการลงทะเบียนหรือไม่)
- ❖ สร้าง Session เพื่อบันทึก Username

## Logout.php

- ❖ ทำลาย Session

# References



- <http://php.net/manual/en/book.session.php>