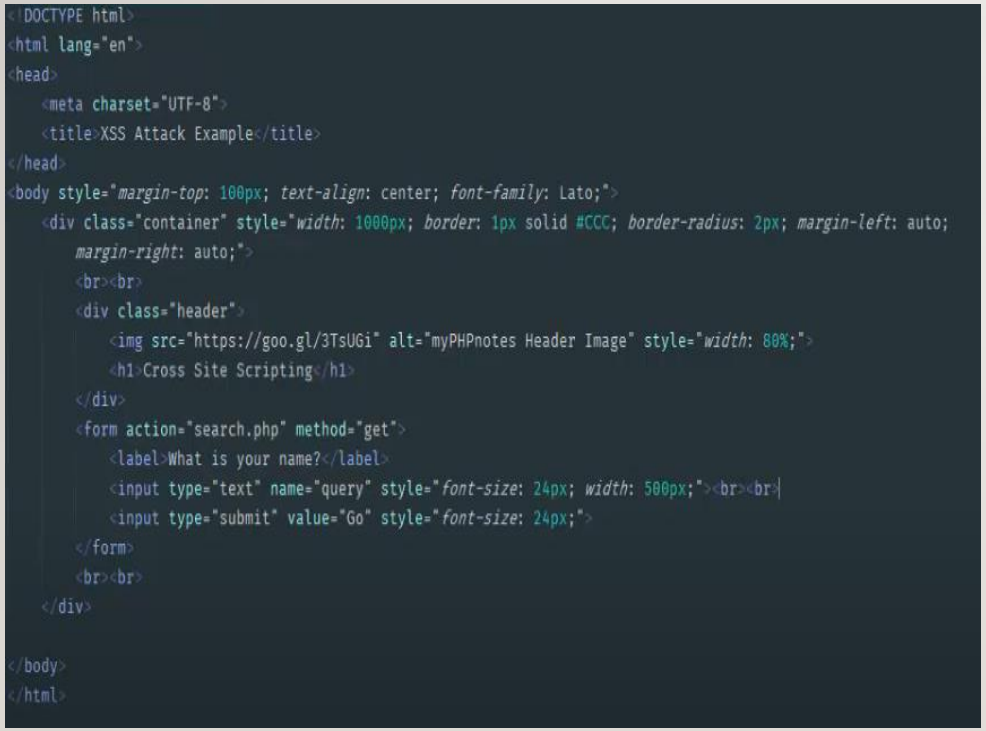
XSS-PROTECT AND SANDBOXING

HOW TO USE XSS-PROTECT?

- Cross-Site Scripting (XSS)
  - A type of injection in which malicious scripts are injected into otherwise benign and trusted websites
  - Occurs when an attacker uses a web application to send malicious code to different end user
- CSS is prevalent to hackers who drive their way via cookies
  - Inputting malicious input using php script variables in order to get confidential data of a user and hacking their computer via their personal accounts

EXAMPLE OF CROSS-SITE SCRIPTING



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>XSS Attack Example</title>
</head>
<body style="margin-top: 100px; text-align: center; font-family: Lato;">
    <div class="container" style="width: 1000px; border: 1px solid #CCC; border-radius: 2px; margin-left: auto;
        margin-right: auto;">
        <br><br>
        <div class="header">
            <img src="https://goo.gl/3TsUGi" alt="myPHPnotes Header Image" style="width: 80%;">
            <h1>Cross Site Scripting</h1>
        </div>
        <form action="search.php" method="get">
            <label>What is your name?</label>
            <input type="text" name="query" style="font-size: 24px; width: 500px;"><br><br>
            <input type="submit" value="Go" style="font-size: 24px;">
        </form>
        <br><br>
    </div>

</body>
</html>
```

- With the picture in the right, we can include malicious input, such as <script>alert(“hello, world”);</script> and <script>location.href=*link*.cookie</script>
- These malicious inputs can cause hackers to get IP addresses easily, and can log in to several accounts that you have for your personal use
- Link to refer:
  - https://www.youtube.com/watch?v=cUdpWTx8o4I

HOW TO PREVENT XSS ATTACKS? (ONE WAY OF PREVENTION)

```php
<?php
header("X-XSS-Protection: 0");

$name = htmlspecialchars($_GET['query'], ENT_QUOTES, 'UTF-8');


echo "Hello, " . $name;
```

- There is a class method called htmlspecialchars, where we can deter any type of scripting language that can capture cookies, data, and other things from a user
  - Convertion of special characters to HTML entities
- In the host page, we can put any scripting language, and what that does is simply state whatever the user put as the input
  - No data will be leaked

SANDBOXING IN PHP

- Runkit-Sandbox
  - Instantiating the Runkit_Sandbox class creates a new thread with its own scope and program stack.
  - Using a set of options passed to the constructor, this environment may be restricted to a subset of what the primary interpreter can do and provide a safer environment for executing user supplied code.
  - Constructor example:
    - Runkit_Sandbox::__construct([array $options]): void

runkit.superglobal

Comma separated list of variables to be treated as superglobals within the sandbox sub-interpreter. These variables will be used in addition to any variables defined internally or through the global runkit.superglobal setting.

SANDBOXING TECHNIQUES IN GITHUB REPOSITORIES

- https://github.com/fregster/PHPSandbox
  - This link provides a great infrastructure to test sandbox in PHP applications
    - Purpose:
      - Allows for running non-trusted PHP from within your main PHP application
      - Allows the main script to continue and indicate fatal errors as it determines in the running code
      - What does it support?
        - Function restrictions
        - Environment obscurification
        - Passing GET and POST parameters
        - Prevents interaction with the parent PHP
- Available in PHP 5.2 and above and PHP CLI

MORE SANDBOXING TECHNIQUES

- shell_exec and backtick operator
  - Executes command via shell and return the complete output as a string
  - It is similar to the getTempVars() function
    - Indicates the user of what type of input they are using, in terms of function calls, class methods, and variable types that they inject in a PHP command
- Since certain web browsers are prone to XSS Protect attacks, they tend to prevent scripting language that forces an output with all the user data
  - This shows how the browser itself is a sandbox (can be tested in a VM using Firefox, since Firefox doesn't have a sandbox as of today)