

Assignment – 4: PHP

이 과제는 총 3 개의 다른 Task 으로 구성되어 있으며, 지금까지 배운 내용을 이용하셔야 합니다.. Task-2 와 Task-3 을 해결하기 위해 Goorm 을 참고할 수 있게, 보조 자료는 만들었습니다. 보조 자료는 “Assignment-4: PHP[Only Study]”라는 이름으로 제공됩니다. 이 보조 자료는 Task-2 와 Task-3 을 해결할 때 도움을 줄 수 있게 설명과 예제들을 포함하고 있으며, `empty()`, `strlen()`, `ord()`, `chr()`, `ctype_lower()`, `ctype_upper()` 함수들과 `.=` 연산자에 대한 내용을 가르칩니다. **보조 자료는 필수는 아니지만, 참고하는 것을 강력히 추천하는 것을 알려드립니다.**

Task 1: Form Handling Implementation on Your Hometown Page

과제 설명:

- 본인의 기존 Hometown 페이지에 PHP 를 이용한 form handling 을 구현하세요.
- Form 을 통해 사용자로부터 입력을 받고, 서버에서 사용자 입력 데이터를 표시하세요.
- Form 에 대한 구체적인 양적 요구사항은 없으나, Form 과 서버 응답이 오류 없이 성공적으로 작동해야 합니다.
- JS 와 CSS 코드에는 수정이 필요 없으며, PHP 파일을 생성하고 HTML 파일에 단 한 줄의 조정만 추가하면 됩니다.

핵심 요구사항:

1. Form Creation:

(Assignment 1~3 요구사항을 이미 했으면은 완료된 상태입니다. Form, CSS, JS 스크립트에 추가 수정이 필요 없습니다.)

- Form 을 디자인하고 Hometown 페이지에 포함하세요.
- 텍스트, 이메일, select, textarea 등 적절한 input type 을 사용해 사용자의 의미 있는 정보를 수집하세요.
- 모든 input 에는 PHP 에서 처리될 수 있도록 적절한 name 속성을 부여하세요.

2. Form Submission:

- Form 의 method 를 POST 로 설정하세요.
- action 속성을 사용하여 Form 데이터를 처리할 PHP 스크립트를 지정하세요.

3. PHP Form Handling:

- Form 데이터를 처리하기 위한 PHP 스크립트를 작성하세요.
- 제출된 데이터를 서버 응답 페이지에 표시하세요.

4. Testing Environment:

- VS Code 에서 PHP 서버를 실행하고 테스트하세요.
- Form 이 정상적으로 작동해야 합니다.
- 서버는 사용자의 입력 데이터를 표시해야 하며, 오류나 비정상 작동이 없어야 합니다.

5. 파일 이름 규칙:

- HTML 파일: studentID_hometown.html
- CSS 파일: studentID_styles.css
- JS 파일: studentID_script.js
- PHP 파일: studentID_back.php
- 이미지 폴더: 이미지가 있을 경우 이곳에 저장하세요.
- 예: 202010921_hometown.html, 202010921_styles.css, 202010921_script.js, 202010921_back.php
- 이 HTML, CSS, JS, PHP 파일들을 studentID_task1 폴더에 저장하세요.

Task 2: Prime Numbers Finder Within Range

과제 설명:

사용자가 정의한 범위 내에서 소수를 찾는 간단한 웹 애플리케이션을 만드세요.

요구사항:

1. Front-end (HTML):

- 프론트엔드 부분을 위한 HTML 파일(studentID_prime.html)을 만드세요.
- 숫자 범위를 정의하기 위해 두 개의 input 필드를 포함한 Form 을 디자인하세요.
- Form 에 백엔드 요청을 보내기 위한 버튼을 추가하세요.

2. Backend (PHP):

- 백엔드 처리를 위한 PHP 스크립트(studentID_prime.php)를 작성하세요.
- Form 으로부터 입력값을 가져오세요.
- 입력받은 데이터를 기반으로 범위 내 모든 소수를 계산하세요.
- 최적화 기법을 구현하면 추가 점수를 받을 수 있습니다(필수는 아님).
- 소수가 없는 경우 사용자에게 알리세요.
- 소수가 있는 경우 소수를 나열하세요.
- 이 작업을 수행하기 위해 새로운 함수들을 선언하세요.
- 서버는 함수에서 반환된 값을 표시해야 합니다.

3. 파일 이름 규칙:

- HTML: studentID_prime.html (예: 202010921_prime.html)
- PHP: studentID_prime.php (예: 202010921_prime.php)
- 이 HTML 및 PHP 파일들을 studentID_task2 폴더에 저장하세요.

예시:

입력:

- 시작 범위: 2
- 종료 범위: 10

출력:

- 범위 내 소수 (2 부터 10): 2, 3, 5, 7

Task 3: Encoding and Decoding with Caesar Cipher

과제 설명:

사용자가 입력한 텍스트와 시프트 값을 받아 인코딩 또는 디코딩을 수행하는 Caesar Cipher 알고리즘을 구현하는 웹 애플리케이션을 만드세요.

요구사항:

1. Front-end (HTML):

- 프론트엔드 부분을 위한 HTML 파일(studentID_cipher.html)을 만드세요.
- Form 을 생성하여 사용자가 다음을 입력할 수 있도록 하세요:
 - 인코딩 또는 디코딩할 텍스트
 - 시프트 값 (1-25)
 - 작업 선택 (인코딩 또는 디코딩)

2. Backend (PHP):

- Form 입력을 처리하고 Caesar Cipher 알고리즘을 실행하기 위한 PHP 스크립트(studentID_cipher.php)를 작성하세요.
- 지정된 시프트 값을 기준으로 텍스트의 문자를 이동시키세요.
- 대문자와 소문자를 따로 처리하세요 (예: 시프트 값이 3 일 경우, A 는 D 로, a 는 d 로 변환).
- 알파벳이 아닌 문자는 변경하지 마세요.
- 인코딩 및 디코딩 작업 모두 처리하세요 (예: 시프트 값이 3 일 경우, a 는 인코딩 시 d 로, 디코딩 시 다시 a 로 변환).
- 결과는 단일 문자열로 출력되어야 하며, 문자가 순차적으로 출력되지 않도록 하세요.
- 도구를 재사용할 수 있도록 프론트엔드 페이지로 돌아갈 수 있어야 합니다.

3. 파일 이름 규칙:

- HTML: studentID_cipher.html (예: 202010921_cipher.html)
- PHP: studentID_cipher.php (예: 202010921_cipher.php)
- 이 HTML 및 PHP 파일들을 studentID_task3 폴더에 저장하세요.

예시 1:

입력:

- 텍스트 입력: PrOgRaMmIng
- 시프트 값 (1-25): 5
- 작업: Encode

출력:

- 원본 텍스트: PrOgRaMmIng
- 시프트 값: 5
- 작업: Encode
- 결과: UwTIWfRrNsl

예시-2:

입력:

- 텍스트 입력: UwTIWfRrNsl
- 시프트 값(1-25): 5
- 작업: Decode

출력:

- 원본 텍스트: UwTIWfRrNsl
- 시프트 값: 5
- 작업: Decode
- 결과: PrOgRaMmIng

최종 제출:

1. 과제를 완료한 후, 각 작업에 대해 아래와 같이 3 개의 서로 다른 폴더를 생성하세요:
 - studentID_task1 (HTML, CSS, JS, PHP 파일 및 이미지 폴더 포함)
 - studentID_task2 (HTML 및 PHP 파일 포함)
 - studentID_task3 (HTML 및 PHP 파일 포함)
2. 새 폴더를 생성하고 이름을 studentID_assignment4 로 지정하세요 (예: 202010921_assignment4).
3. 위에서 생성한 studentID_task1, studentID_task2, studentID_task3 폴더를 studentID_assignment4 폴더에 넣으세요.
4. studentID_assignment4 폴더를 압축하여 studentID_assignment4.zip 으로 저장하세요 (예: 202010921_assignment4.zip).
5. e-class 에 zip 파일을 제출하세요.
6. 제출 마감: 2024 년 12 월 12 일, 목요일, 11:59:59 PM.

제출 전 체크리스트:

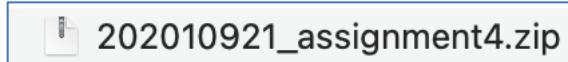
- Goorm 의 보조 자료를 참고하여 관련 함수들과 내용을 학습했는지 확인하세요(필수 아님).
- 모든 파일 이름과 폴더 이름이 올바르게 지정되었는지 확인하세요(필수).
- 모든 작업이 VS Code 의 PHP 서버에서 오류 없이 작동하는지 테스트하세요(필수).

- zip 파일을 12 월 12 일까지 e-class 에 제출하세요.

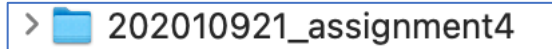
문의 사항은 카카오톡이나 sercanyesilkoy@hufs.ac.kr 로 **조교에게만 연락하세요**

Example of naming for files:

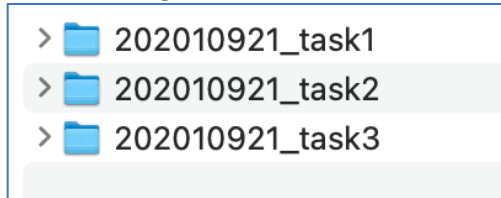
Zip file:



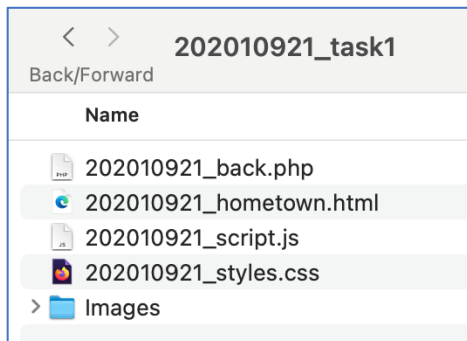
Inside of zip:



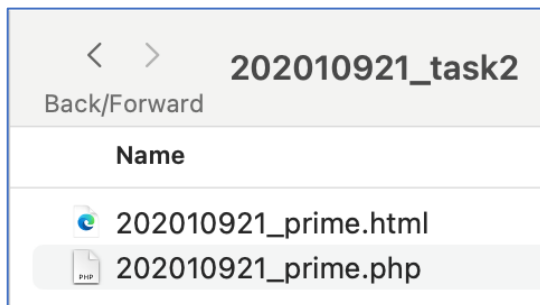
Inside of assignment4 folder:



Inside of task1 folder:



Inside of task2 folder:



Inside of task3 folder:

