

**JUNIOR-STUDY 과제 – Web3**

**제출일자: 2022.04.30**

**학과: 생활과학계열**

**학번: 202121001**

**이름: 김한솔**

**목차**

**1. 실습 과제**

**2. Php에서 ‘와 “의 차이점 조사**

**3. 끝말잇기 게임 구현**

**4. 숫자야구 게임 구현**

**5. 완성된 회원가입 폼에서 보낸 정보 출력**

**6. Cookie와 Session 조사**

**1. 실습 과제**

**실습#1**

PHP를 이용해 구구단 만들기

|  |
| --- |
| <http://junior.catsecurity.net/~casol0219/실습1구구단.php> |

텍스트이(가) 표시된 사진

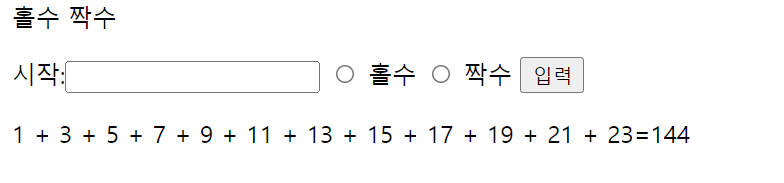
자동 생성된 설명

|  |
| --- |
| <html>  <head>  <title>실습과제1#구구단</title>  </head>  <body>  <form method="post" action="">  입력:<input type="text" name="dan"/>  <input type="submit" value="전송"/>  </form>  </body>  </html>  <?php  $j=1;  do{  echo $\_POST['dan']."X".$j."=".$\_POST['dan'] \* $j."<br>";  $j++;  }while($j<10);  ?> |

**실습#2**

N의 자연수가 주어질 때, 이들 중 홀수 혹은 짝수를 선택해, 해당 자연수들을 모두 골라 그 합을 구하는 php를 작성하라

|  |
| --- |
| <http://junior.catsecurity.net/~casol0219/실습2.php> |



|  |
| --- |
| <html>  <head>  <title>실습과제2</title>  </head>  <body>  <p>홀수 짝수</p>    <form method="post" action="">  시작:<input type="text" name="number"/>    <input type="radio" name="n" value="홀수"/> 홀수  <input type="radio" name="n" value="짝수"/> 짝수    <input type="submit" value="입력"/>  </form>  </body>  </html>  <?php  $sum1=0;    if($\_POST['n']=="짝수"){  for($i=1;$i<=$\_POST['number'];$i++){  if($i%2==0){  if($i==$\_POST['number']){  echo $i."=";  $sum1+=$i;  }  else{  echo $i." + ";  $sum1+=$i;  }  }  }  }  else if($\_POST['n']=="홀수"){  for($i=1;$i<=$\_POST['number'];$i++){  if($i%2!=0){  if($i==$\_POST['number']){  echo $i."=";  $sum1+=$i;  }  else{  echo $i." + ";  $sum1+=$i;  }  }  }  }  echo $sum1;  ?> |

**2. Php에서 ‘와 “의 차이점 조사**

문자열 내부에 변수나 개행 문자가 존재하는 경우를 통해 차이를 확인해보았다.

|  |  |
| --- | --- |
| **“ 를 사용했을 때** | |
| **코드** | **실행 결과** |
| **텍스트이(가) 표시된 사진  자동 생성된 설명** |  |

|  |  |
| --- | --- |
| **‘ 를 사용했을 때** | |
| **코드** | **실행 결과** |
| **텍스트이(가) 표시된 사진  자동 생성된 설명** |  |

-큰따옴표는 문자열 내부의 변수를 치환해서 보여주고 \n을 개행 문자로 변환해서 보여주는 반면 작은따옴표는 문자열을 그대로 출력해준다.

-이러한 이유로 PHP에서 큰따옴표보다 작은따옴표의 처리속도가 더 빠르지만 그 차이는 미미하다.

**3. 끝말잇기 게임 구현**

|  |
| --- |
|  |

**6. Cookie와 Session 조사**

**<Cookie>**

사용자가 어떤 웹 사이트를 방문할 경우, 그 사이트가 사용하고 있는 서버에서 사용자의 컴퓨터에 저장하는 클라이언트의 상태 정보 파일이다. 필요시 정보를 참조하거나 재사용할 수 있다.

**<1> Cookie의 특징**

-클라이언트에 300개의 쿠키를 저장할 수 있다

-하나의 도메인 당 20개의 쿠키를 가질 수 있다

-하나의 쿠키는 4KB까지 저장 가능하다

-이름, 값, 유효 시간, 도메인, 경로로 구성되어있다.

-브라우저가 종료되어도 유효 시간이 만료되지 않았다면 계속 클라이언트에서 보관된다

**<2> Cookie의 동작 방식**

(1) 클라이언트가 페이지를 요청 (=사용자가 웹사이트 접속)

(2) 웹 서버에서 쿠키를 생성

(3) HTTP 화면을 돌려줄 때 생성한 쿠키에 정보를 담아 클라이언트에게 돌려줌

(4) 넘겨받은 쿠키는 클라이언트가 가지고 있다가 다시 서버에 요청할 때 쿠키도 함께 전송함

**<3> Cookie 사용 예시**

-쇼핑몰 장바구니 기능

-방문했던 사이트 재방문시 아이디와 비밀번호 자동 입력

-팝업창을 통해 “오늘 이 창을 다시 보지 않기” 체크

**<Session>**

**<1> Session의 특징**

-사용자에 대한 정보를 브라우저에 저장하는 쿠키와 달리 세션은 사용자에 대한 정보를 서버 측에서 관리한다 (웹 서버에 저장되는 쿠키 = 세션 쿠키)

-각 클라이언트에게 고유한 세션 ID를 부여하여 클라이언트를 구분한다.

-세션 ID로 클라이언트를 구분하기 때문에 클라이언트의 요구에 맞는 서비스를 제공할 수 있다

-웹 브라우저가 서버에 접속해서 브라우저를 종료할 때까지 인증상태를 유지한다

(접속 시간에 제한을 두어 일정 시간 응답이 없다면 정보가 유지되지 않게 설정할 수 있다)

-사용자에 대한 정보를 서버에서 관리하기 때문에 보안면에서 쿠키보다 우수하다

-사용자가 많아질수록 서버 메모리를 많이 차지하게 된다

-서버 메모리가 허용하는 한 저장 데이터에 제한이 없다

-브라우저가 종료되었거나 서버에서 세션을 삭제했을 때 삭제된다

**<2> Session의 동작 방식**

(1) 클라이언트가 페이지를 요청 (=사용자가 웹사이트 접속)

(2) 서버는 생성한 쿠키에서 세선 ID를 확인한 후 없으면 세션 ID를 생성해 클라이언트에게 쿠키와 함께 돌려줌

(3) 클라이언트는 서버에 요청할 때 쿠키와 함께 세션 ID를 서버에 전달해서 요청

(4) 서버는 세션 ID를 전달받아 클라이언트 정보를 가지고 서버 요청을 처리하여 클라이언트에게 응답

**<3> Session 사용 예시**

-화면이 이동해도 로그아웃 되지 않고 로그인 유지