kmunity

20113315 이형준

20113316 임병준

홈페이지 디자인

|  |  |
| --- | --- |
| **div id = header // top\_login.php 계속 include하여사용** | |
| **div id = menu // top\_menu.php 계속 include 하여 사용** | |
| **div id = col1**  **include 하여사용** | **div id = col2** |
|  |

**특이사항**

-header , left\_menu , top menue include 동일

-모든 것은 div로 처리. 횡으로 div 쌓을땐 css로 float:left

후 div id 는 ex)list\_search1 , list\_search2, list\_search3 이런식으로 쌓는다.

clear both; 로 횡 으로 쌓는 div마무리

* li 리스트 에서 css display:inline 으로 횡으로 나열할수있다.

**<파일 구조>**

**db / 페이지 ( html을 포함한 php) / 기능구현(php)**

lib 파일

* php파일들을 include 해서 사용한다.

dbconn.php : db에 접속하고 선택하는 것

top\_login : 좌측 상단 logo , moto 및 우측상단 로그인 / 회원가입

top\_menu : 낙서장 / 가입인사 / 자유게시판. 등 메뉴

left\_menu: 왼쪽 메뉴바

회원가입처리

**파일 목록 구성**

|  |  |  |
| --- | --- | --- |
|  | 파일명 | 기능명 |
| db | member.sql | 회원 정보 db |
| 페이지 | member\_form.php | 회원가입 페이지 |
| 기능구현(php) | insert.php | db에 회원 가입 레코드 추가 |
| check\_id.php | 아이디 중복확인 |
| check\_nick.php |  |

1. 데이터 베이스 설계
2. Member.sql

-필드명 : id pass name nick hp email regist\_day level

1. 회원가입 페이지 생성
2. 중복확인기능

* window.open 을 이용한 post 방식으로 데이터 넘기기.

check\_id로 $id정보를넘긴다.

1. 사용자가 입력한 정보를 db에 저장.(중복확인 기능)

-member\_form.php의 <div id=”content”>안의 저장하기 버튼클릭시 function check\_input() 실행되서 submit();

-insert.php에서 변수들을 $\_POST[]로 받아서 $sql에 레코드 삽입 명령을 입력.

-mysql\_query($sql, $connect); 로$sql 에 저장된 명령 실행

로그인 / 로그아웃

**파일 목록 구성**

|  |  |  |
| --- | --- | --- |
|  | 파일명 | 기능명 |
| 페이지 | login\_form.php | 로그인 페이지 |
| member\_form\_confirm.php | 회원정보확인 |
| member\_form\_modify.php | 회원정보수정 |
| 기능구현(php) | login.php | 로그인 처리( sql 연동) |
| logout.php | 로그아웃 처리 |
| modify.php | 회원정보 수정 |

1. 로그인 페이지

- login\_form.php

<input type="image" src="../img/login\_button.gif">

login 이미지를 누르면 post방식으로 action=”login.php”에 변수들이 전달된다.

($id , $pass )

1. 로그인 처리
2. 알고리즘 : id와 pass를 넘겨받아서 php파일에서 id를 db에서 검색 후 있으면 비밀번호를 대조한다. 일치하면 세션 변수 등록.
3. login.php

$result = mysql\_query($sql, $connect);

// $sql에 입력된 명령어를 실행시킨다. 성공:ture,실행결과값.. 실패:false

$num\_match = mysql\_num\_rows($result);

// 명령어 실행결과값에서 레코드 개수를 알려준다 성공:레코드의 개수 실패:false

$row = mysql\_fetch\_array($result);

//명령어 실행결과값에서 레코드 정보를 배열로 가져온다. 성공:배열형태의 레코드 실패:f

1. 세션변수

-전역변수 $\_SESSION[ ] 로 등록해서 다른페이지에서 로그인 여부 확인가능.

ex) $\_SESSION[‘userid’] = $userid;

1. 로그아웃 처리

unset($\_SEESION[‘userid’]);

// 로그인할 때 저장한 세션변수를 해제시켜 준다. 그리고 메인화면으로돌아간다.

1. 회원정보수정

-$hp=explode(“-”,$row[hp]);

$hp1=$hp[0]; $hp2=$hp[1]; $hp3=$hp[2];

//$row[hp] 를 받아와서 “–“ 기준으로 잘라 배열에 넣어둔다.

**Git Hub**

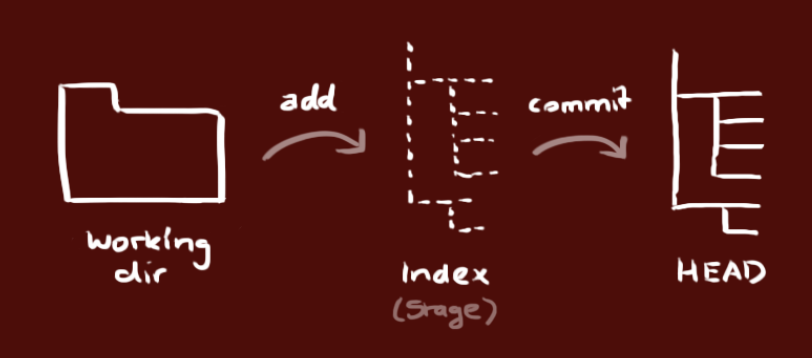
**-구성요소**

1.원격저장소

- 인터넷 git hub를 의미한다.

2.로컬저장소

1. Working directory
2. Index (준비영역 역할)
3. HEAD (최종 확정본)



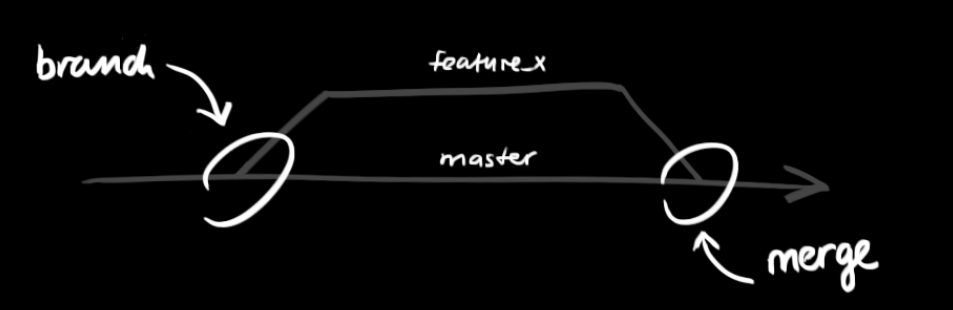
**-작업하는법** (이전에 clone한 폴더는 지우고 새로 clone하고 시작해야 된다)

|  |  |  |
| --- | --- | --- |
| 순서 | 작업명 | 명령어 |
| 1 | 새로운 저장소 만들기 | git init |
| 2 | 원격저장소->로컬저장소 | git clone 주소 |
|  | clone 후 해당 디렉토리로 이동한다. | |
| 3 | 변경된 파일을 index에 추가 | git add 파일이름 |
| 4 | 변경 내용 확정 | git commit –m “확정본에대한 설명” |
| 현재까지는 로컬 저장소에 있는 상태이다. 상태 확인 명령어 : git status | | |
| 5 | 로컬 -> 원격저장소 올리기 | git push origin 가지이름(master) |

**-Branch**

1. 원격 저장소 생성시 master 가지가 생성됨.

2. 다른 가지를 이용해, 개발을 진행하고 후에 개발 완료시 master로 돌아와 병합해야함.

****

|  |  |  |
| --- | --- | --- |
|  | 작업명 | 명령어 |
|  | 가지를 생성하고 갈아탄다. | git checkout –b 파일명 |
|  | Master 가지로 돌아온다 | git checkout master |
|  | 가지를 삭제한다 | git branch –d 파일명 |

**-갱신 및 병합**

|  |  |
| --- | --- |
| 작업명 | 명령어 |
| 로컬저장소를 원격저장소에 맞춰 갱신한다  ( 원격 -> 로컬) | git pull |
|  |  |
|  |  |

게시판

**파일 목록 구성**

|  |  |  |
| --- | --- | --- |
|  | 파일명 | 기능명 |
| db | board.sql | 게시판 내용 db |
| 페이지(html포함) | list.php | 글 목록 페이지 |
| write\_form.php | 글쓰기 페이지(수정포함) |
| view.php | 글 내용 보기 페이지 |
| 기능구현(php) | insert.php | 글 저장 ( 수정 후 저장) |
| delete.php | 글 삭제 |
| 데이터 저장 | data | 이미지 업로드 폴더 |

**특이사항**

1. 로그인 해야만 글쓰기가능
2. 닉네임

**제작과정** :

1. 데이터 베이스 설계

1) board.sql

-필드명 : id pass name nick hp email regist\_day level

2. 글목록 확인 ( list.php)

1) 기능구현

a. 검색기능

검색 버튼(input type=”image”) 클릭시 action=”list.php?mode=search”

으로 select name=”find” 변수와 같이 넘어간다.

$sql=”select \*from greet where $find like ‘’%$search%’ order by num desc”

필드명인 $find에서 $search가 포함된 레코드를 찾은후 num(일련번호) 순으로 내림차순 정렬한다

$total\_record //전체 글의 수

$total\_page //전체 페이지 수

$start //해당 페이지의 시작 레코드 번호

//

$scale //한페이지에 보이는 레코드 개수

for문을 10번씩돌려서 레코드를 하나씩 가져온다.

b. List -> view

<a href="view.php?num=<?=$item\_num?>&page=<?=$page?>"><?= $item\_subject ?>

글 제목을 클릭시 get방식으로 num (게시물 순서)과 page 를 view.php로 넘긴다.

view.php 에서 num을 이용하여 db에서 레코드를 추출해 내고

출력한다.

1. view -> modify

get 방식으로 num과 page를 넘긴다.

modify.php 에서 목록보기를 클릭할경우 해당페이지의 목록을 보여줘야 하기떄문에 페이지 번호(page)도 보낸다.

1. view -> delete.
2. 글쓰기

* insert.php에 subject, content, html\_ok 가 넘어간다

HTML쓰기 선택시 insert.php로 $html\_ok 에 y가 전달

* insert.php

글 저장기능 / 글 수정기능..

**이미지를 첨부할 수 있는 글쓰기 페이지 생성**.

1. **요구사항**
2. JPG, GIF 포맷만 업로드 가능.
3. 크기가 500KB이하인 이미지 파일만 업로드 가능.
4. **db테이블 설계**

file\_name : 업로드 하는 이미지의 실제 파일명

file\_copied : 이미지가 data폴더에 저장되는 파일명(작성일과 시간으로 새로 붙인이름명)

1. **게시판의 글쓰기 페이지( write\_form.php )**

이미지 파일1 찾아보기 상자

<input type:”file” name=”upfile[]”>

이미지 삭제하기

<input type=”checkbox” name=”del\_file[]” value=”0”>

* insert.php로 배열변수들이 전달된다 . 451p참조

|  |  |
| --- | --- |
| 배열 | 설명 |
| $\_FILS[“upfile”][“name”] | 업로드 하는 실제 파일명 |
| $\_FILS[“upfile”][“tmp\_name”] | 서버에 저장되는 임시 파일명 |
| $\_FILS[“upfile”][“type”] | 업로드 파일 형식 |
| $\_FILS[“upfile”][“size”] | 업로드 파일 크기 |
| $\_FILS[“upfile”][“error”] | 에러 발생 확인 |

1. **글 저장하기**

파일을 data폴더에 업로드하기.

$files = $\_FILS[“upfile”]; //3차원배열인 $\_FILES를 변수에 저장

$count=count($files[“name”]); // 업로드할 파일의 개수를 변수count에 저장.

$upload\_dir=’./data/”;