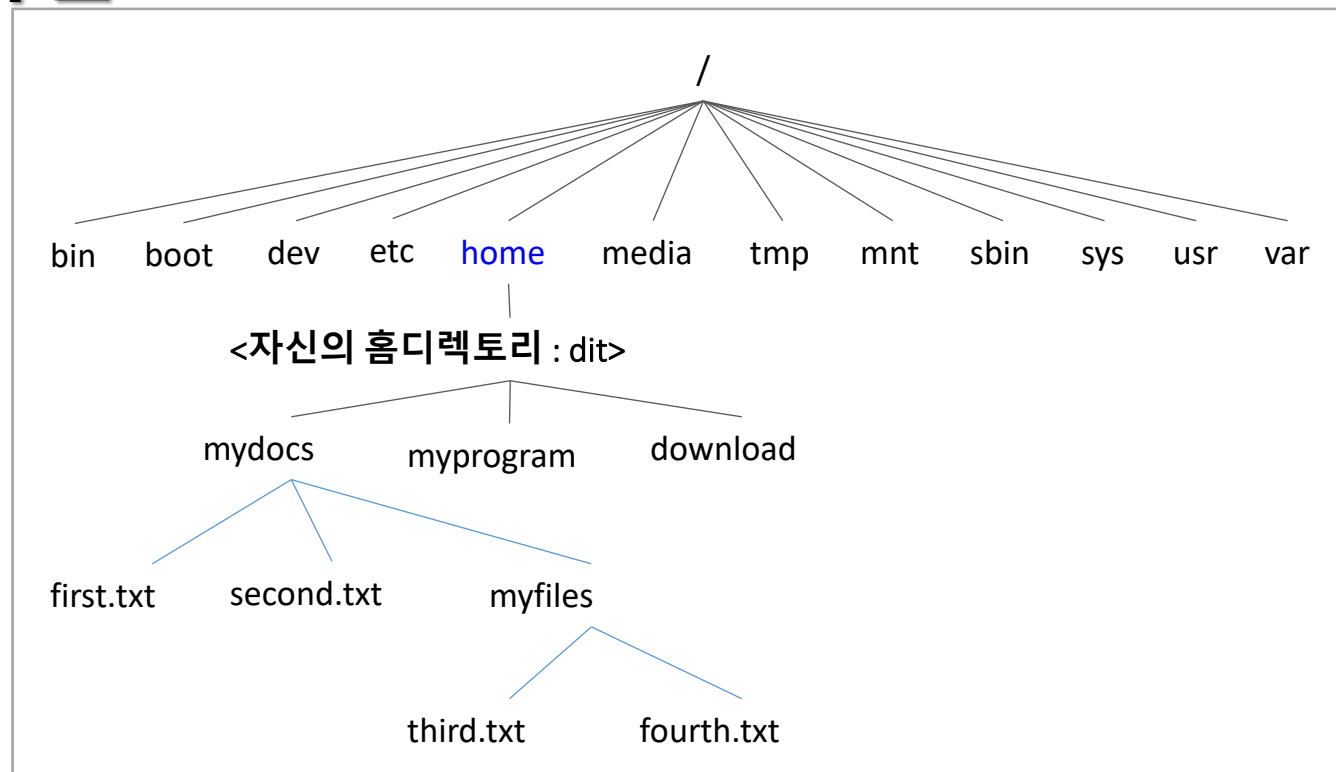


vi, linux text editor

이전 시간 복습



- 자신의 홈디렉토리에 있는 파일들을 모두 지우시오.
- 다음과 같이 디렉토리와 파일을 작성하시오.
- 현재 위치에서 다른 디렉토리로 이동하는 것을 연습하시오.
 - fourth.txt -> nana.txt 로 이름 변경
 - second.txt 파일을 복사하여 download 디렉토리에 저장하기 등

vi 소개

- Linux(UNIX) 텍스트 에디터
- 세상에서 가장 가볍고 강력한 편집기능을 제공하는 도구 중의 하나
- 기본 키보드만 사용하며 편집
(마우스, 화살표 키 사용할 필요 없음)
- 단순 학습보다는 직접 경험을 쌓는 것이 중요

vi 실행

Shell 실행 명령	내용
\$ vi	vi 실행 (편집 후 저장 가능) (vim (vi improved)이 실행되는 os도 있음)
\$ vi <i>filename</i>	File이 존재하면 해당 파일 열고 편집 File이 없으면 빈 내용으로 편집 시작

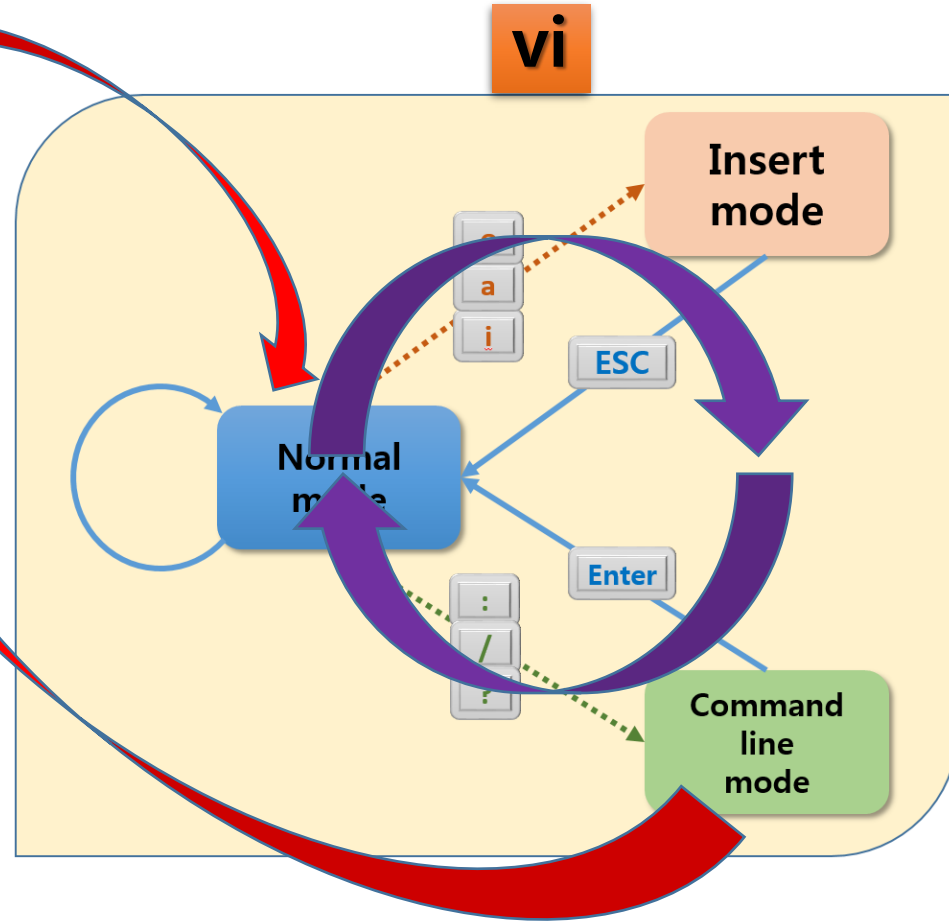
vi, 실행 → 편집 → 종료

shell

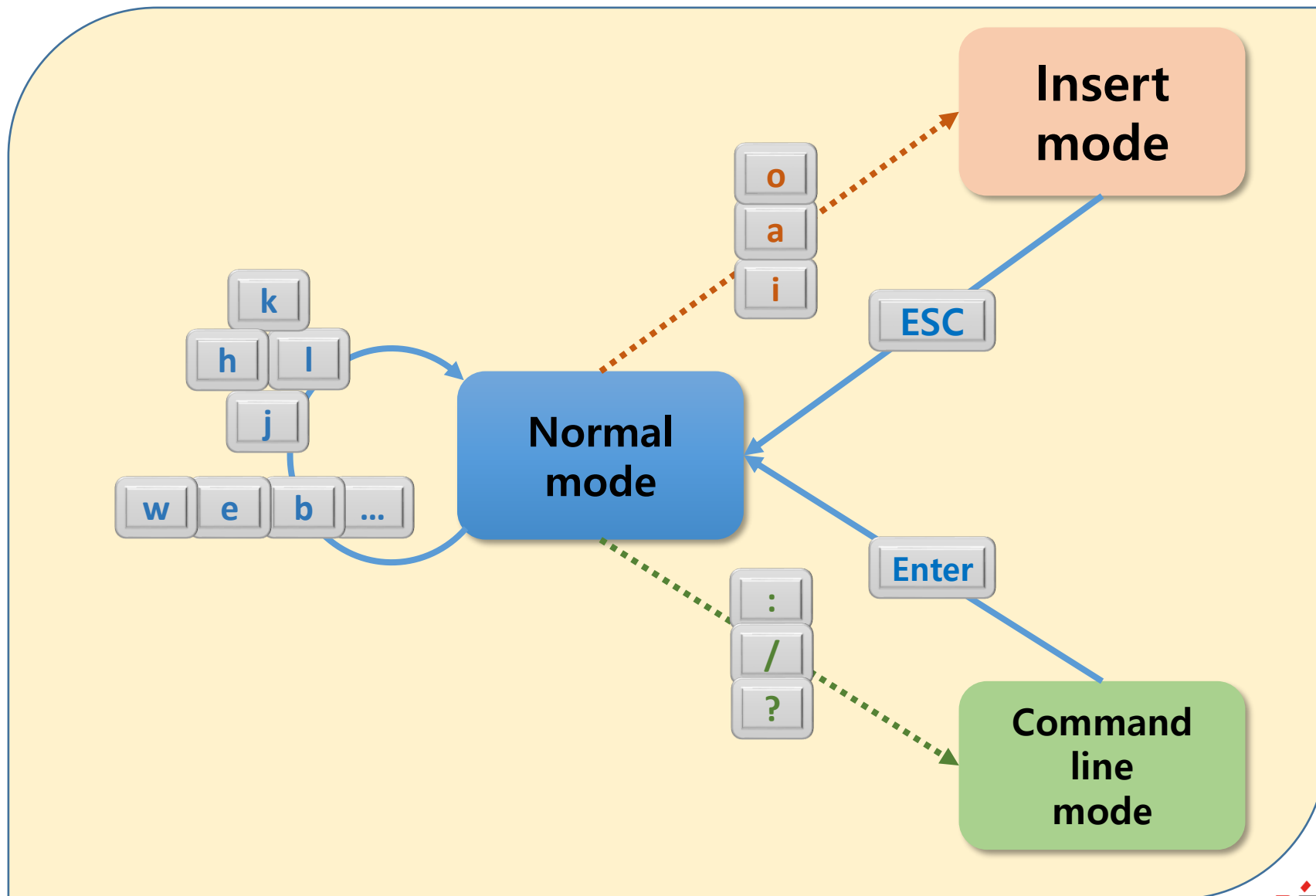
```
$ vi filename  
$
```

vi 편집 작업은 3개의 모드를 오가며 진행

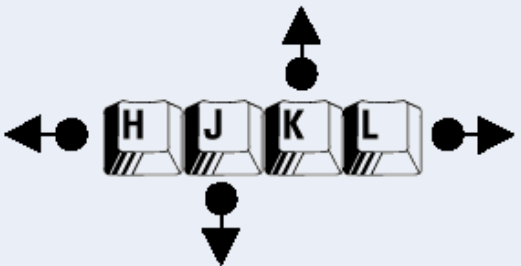
vi



vi mode 간 이동 명령 미리 보기



vi 커서 이동 연습 (따라 하기 ②)

명령	내용
\$ vi first.txt	first.txt 편집 시작 (vi Normal mode 로 진입)
	Normal Mode 에서 h, j, k, l 키 만으로 커서를 이동할 수 있다. 👉 익숙해질 때까지 충분히 연습
:q	Command mode 로 이동(:)하여 종료(q)

Tip!

:set nu

줄 번호 표시하기 설정

:set nonu

줄 번호 표시하기 해제

Normal mode → Insert mode (따라 하기 ③)

명령	내용
i	커서 위치 문자 앞에서 입력 시작
I	커서 줄의 맨 앞으로 이동하여 입력 시작
a	커서 위치 문자 뒤에서 입력 시작
A	커서 줄의 맨 뒤로 이동하여 입력 시작
o	커서 아래에 새 줄을 삽입 하고 입력 시작
O	커서 위에 새 줄을 삽입 하고 입력 시작

Normal mode **커서 이동 2**

명령	내용
w, b	단어 단위 이동: (w : 다음 단어, b : 이전 단어)
e	다음 단어의 끝 글자로 이동
^	현재 줄의 맨 앞으로 이동 (빈 칸이 아닌 첫 글자로 이동)
0, \$	현재 줄의 맨 앞(0) 또는 맨 뒤(\$)로 이동 (빈 칸 포함)
<i>n</i>G	<i>n</i> 번 줄로 이동, G : 파일의 끝으로 이동
Ctl-E Ctl-Y	한 줄씩 위 아래로 화면 스크롤
Ctl-D Ctl-U	반 화면씩 위 아래로 화면 스크롤
J	현재 줄 뒤에 다음 줄을 이어서 연결한다.

Normal mode 삭제 명령

명령	내용
x	커서 위치 문자 삭제
X	커서 위치 앞 문자 삭제
D	커서 위치 문자부터 그 줄의 끝까지 삭제
dw	단어 삭제 (커서 위치 문자부터 다음 단어 앞의 공백까지 삭제)
dd	커서 위치 줄 삭제
C	커서 위치 문자부터 줄 끝까지 지우고 Insert mode
cc	커서 위치 줄 전체 지우고 Insert mode
u	마지막 명령 취소 (반복 적용 가능)
<i>n</i>	반복할 횟수(<i>n</i>)과 함께 위의 명령을 사용하면 반복 적용

Normal mode 복사/이동

명령	내용
yy, Y	커서 위치 줄 복사 (클립보드에 복사)
p	클립보드 내용을 커서 위치 줄 아래에 삽입
P	클립보드 내용을 커서 위치 줄 위에 삽입
복사	yy, Y로 복사하고 p, P로 추가하기
이동	dd로 삭제하고 원하는 위치로 이동 후 p, P로 추가하기

* 복사 및 이동 작업은 처리 대상(줄 또는 단어)에 따라 p, P 적용시 다르게 적용된다.

마지막 수정명령 반복

.	편집 버퍼를 수정(삭제, 추가, 복사 등) 마지막 명령 반복
----------	-----------------------------------

Command line mode 기본 명령

명령	내용
:w	편집 중인 파일명으로 저장
:w <i>file</i>	편집 중인 파일을 <i>file</i> 로 별도 저장.(현재 파일은 유지됨)
:q	vi 종료. (편집 파일이 저장되지 않았으면 경고 후 종료되지 않음)
:q!	vi 강제 종료. (저장되지 않았어도 강제로 종료됨)
:o <i>file</i>	<i>file</i> 열기. (현재 파일이 저장되지 않았으면 경고 후 취소)
:r <i>file</i>	현재 줄 다음에 <i>file</i> 내용을 읽어 삽입
:w >> <i>file</i>	현재 편집 중인 파일 내용을 <i>file</i> 뒤에 추가함.
:wq	현재 파일 저장하고 종료

검색 (Normal mode)

명령	내용
<i>/rexp</i>	지정한 패턴(예: <i>rexp</i>)을 앞으로 검색
<i>/</i>	이전에 지정된 검색 패턴을 앞으로 검색
<i>?rexp</i>	지정한 패턴(예: <i>rexp</i>)을 뒤로 검색
<i>?</i>	이전에 지정된 검색 패턴을 뒤로 검색
n	이전의 <i>/</i> 또는 <i>?</i> 명령과 <u>같은</u> 방향 검색 반복
N	이전의 <i>/</i> 또는 <i>?</i> 명령과 <u>반대</u> 방향 검색 반복

치환 (Command mode)

명령	내용
<code>:s/<i>src</i>/<i>rep</i>/</code>	<u>현재 줄</u> 에서 패턴 " <i>src</i> "를 찾아 패턴 " <i>rep</i> "로 치환
<code>:5s/<i>src</i>/<i>rep</i>/</code>	<u>5번 줄</u> 에서 패턴 " <i>src</i> "를 찾아 패턴 " <i>rep</i> "로 치환
<code>:1,5s/<i>src</i>/<i>rep</i>/</code>	<u>1~5번 줄</u> 에서 패턴 " <i>src</i> "를 찾아 패턴 " <i>rep</i> "로 치환
<code>:%s/<i>src</i>/<i>rep</i>/</code>	<u>모든 줄</u> 에서 패턴 " <i>src</i> "를 찾아 패턴 " <i>rep</i> "로 치환

Normal mode

<code>rc</code>	커서 위치의 문자를 ' <i>c</i> '로 치환 (mode 전환 없음)
<code>~</code>	커서 위치의 문자 대소 변환 (변환 후 커서는 한 칸 자동으로 이동)

Command mode :set 변수 일부

명령	내용
:set nu/nonu	줄 번호 표시 / 해제
:set ts=<i>n</i>	Tab stop 크기를 <i>n</i> 으로 설정
:set sw=<i>n</i>	Shift width 크기를 <i>n</i> 으로 설정 (:set si 사용 시 적용)
:set ai / noai	Auto indent 설정/해제 (윗 줄의 들여쓰기 적용)
:set si / nosi	Smart indent 설정/해제. if, while, switch 및 블록 설정 등 뒤에 자동으로 들여쓰기를 해준다.
:set ruler	화면 우측 하단에 커서 위치(행, 열), 파일에서의 위치를 %로 표시
:set key=<i>pass</i> :set key=	파일 저장 시 암호 설정 (열 때도 암호 필요함) 암호 설정 해제

명령어

- **cat**

- 단순히 파일의 내용을 화면에 출력
- 2개 이상의 파일을 연결하여 출력할 때 편리

```
CECTX\std60@std60:~$ vi f1
CECTX\std60@std60:~$ vi f2
CECTX\std60@std60:~$ cat f1
first
CECTX\std60@std60:~$ cat f2
second
CECTX\std60@std60:~$ cat f1 f2
first
second
CECTX\std60@std60:~$
```

- **more**

- 긴 파일을 읽을 때 사용

```
pi@linux:~/test $ more test.txt
ECMAScript 2015 (ES6) and beyond

Node.js is built against modern versions of V8. By keeping up-to-date with the latest
releases of this engine, we ensure new features from the JavaScript ECMA-262 specifica
tion are brought to Node.js developers in a timely manner, as well as continued perfor
mance and stability improvements.

All ECMAScript 2015 (ES6) features are split into three groups for shipping, staged, a
nd in progress features:

All shipping features, which V8 considers stable, are turned on by default on Node.js
and do NOT require any kind of runtime flag.
Staged features, which are almost-completed features that are not considered stable by
the V8 team, require a runtime flag: --harmony.
In progress features can be activated individually by their respective harmony flag, a
lthough this is highly discouraged unless for testing purposes. Note: these flags are
exposed by V8 and will potentially change without any deprecation notice.
Which features ship with which Node.js version by default?

The website node.green provides an excellent overview over supported ECMAScript featur
es in various versions of Node.js, based on kangax's compat-table.

--More-- (36%)
```


실습

Yesterday
All my troubles seemed so far away
Now it looks as though they're here to stay
Oh, I believe in yesterday
Suddenly
I'm not half the man I used to be
There's a shadow hanging over me
Oh, yesterday came suddenly

Why she had to go,
I don't know She wouldn't say
I said something wrong
Now I long for yesterday

Yesterday
Love was such an easy game to play
Now I need a place to hide away
Oh, I believe in yesterday

Why she had to go,
I don't know She wouldn't say
I said something wrong
Now I long for yesterday

Yesterday
Love was such an easy game to play
Now I need a place to hide away
Oh, I believe in yesterday

yesterday.txt

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Insert title here</title>
6 </head>
7 <body>
8     <h2>DB에 로그인 데이터 입력</h2>
9
10 <form action="insertDBPro.jsp" method="post">
11     아이디 : <input type="text" name="id"><br>
12     암호 : <input type="password" name="pwd"><br>
13     <input type="submit" value="확인">
14     <input type="reset" name="재설정">
15 </form>
16 </body>
17 </html>
```

form.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Insert title here</title>
6 </head>
7 <body>
8   <h2>회원가입</h2>
9   <hr>
10 <form action="mRegisterOk.jsp" method="post">
11   <label for="1">아이디 : </label>
12   <input type="text" id="1" name="id"> <br>
13
14   <label for="3">이름 : </label>
15   <input type="text" id="3" name="name"> <br>
16
17   <label for="5">연락처 : </label>
18   <input type="text" id="5" name="tel1" size="3">-
19   <input type="text" name="tel2">-
20   <input type="text" name="tel3">
21   <br>
22
23 <fieldset>
24   <legend>취미생활</legend>
25   <input type="checkbox" name="hobby" value="축구"> 축구
26   <input type="checkbox" name="hobby" value="영화감상"> 영화감상
27 </fieldset>
28
29   <input type="submit" value="입력완료">
30   <input type="reset" value="다시입력">
31 </form>
32 </body>
33 </html>
```

member.html

실습

서시

윤동주

죽는 날까지 하늘을 우러러
한 점 부끄럼이 없기를,
잎새에 이는 바람에도
나는 괴로워했다.
별을 노래하는 마음으로
모든 죽어가는 것을 사랑해야지,
그리고 나한테 주어진 길을
걸어가야겠다.

오늘 밤에도 별이 바람에 스치운다.

poem.txt

- 5줄, 9줄 지우기
- 별을 달로 치환하기
- 마지막 라인을 복사하여 삽입하기
- form.html 파일을 여기에 삽입하여 보이기
- 파일의 암호 설정하기 등

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<div class="container">
  <h2>DB에 로그인 데이터 입력</h2>

  <form action="insertDBPro.jsp" method="post">
    <div class="form-group">
      아이디 : <input class="form-control" type="text" name="id"><br>
    </div>
    <div class="form-group">
      이름 : <input class="form-control" type="text" name="name"><br>
    </div>
    <div class="form-group">
      암호 : <input class="form-control" type="password" name="pwd"><br>
    </div>
    <input class="btn btn-primary" type="submit" value="확인">
    <input class="btn btn-primary" type="reset" name="재설정">
  </form>
</div>
</body>
</html>
```

확인 학습

- <https://bit.ly/3zUFPtc>