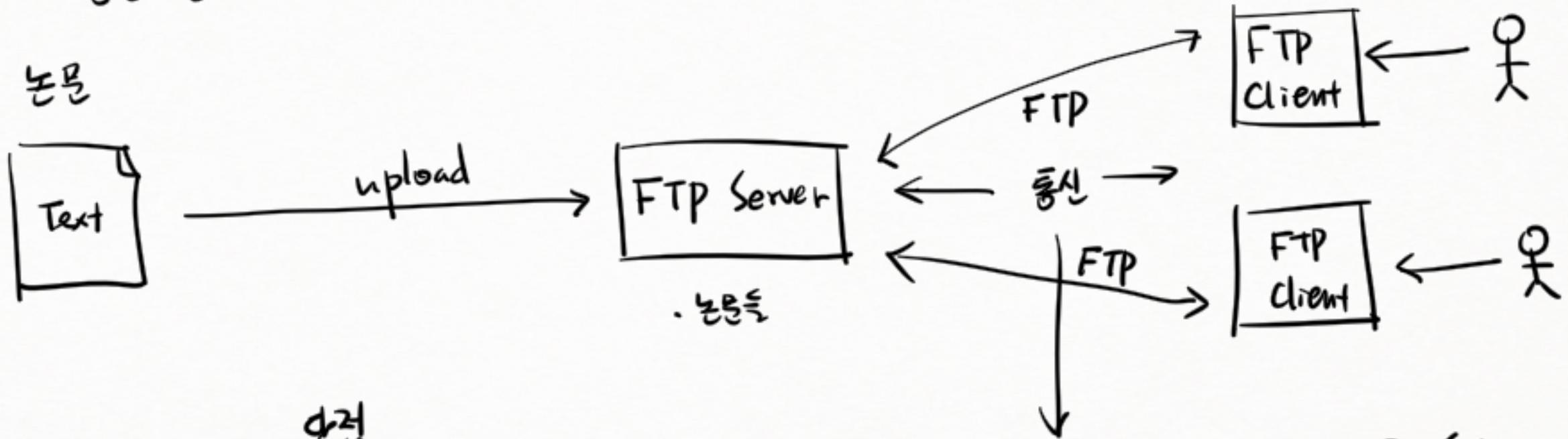


* HTML, HTTP

① HTTP, HTML 등장 전



단점

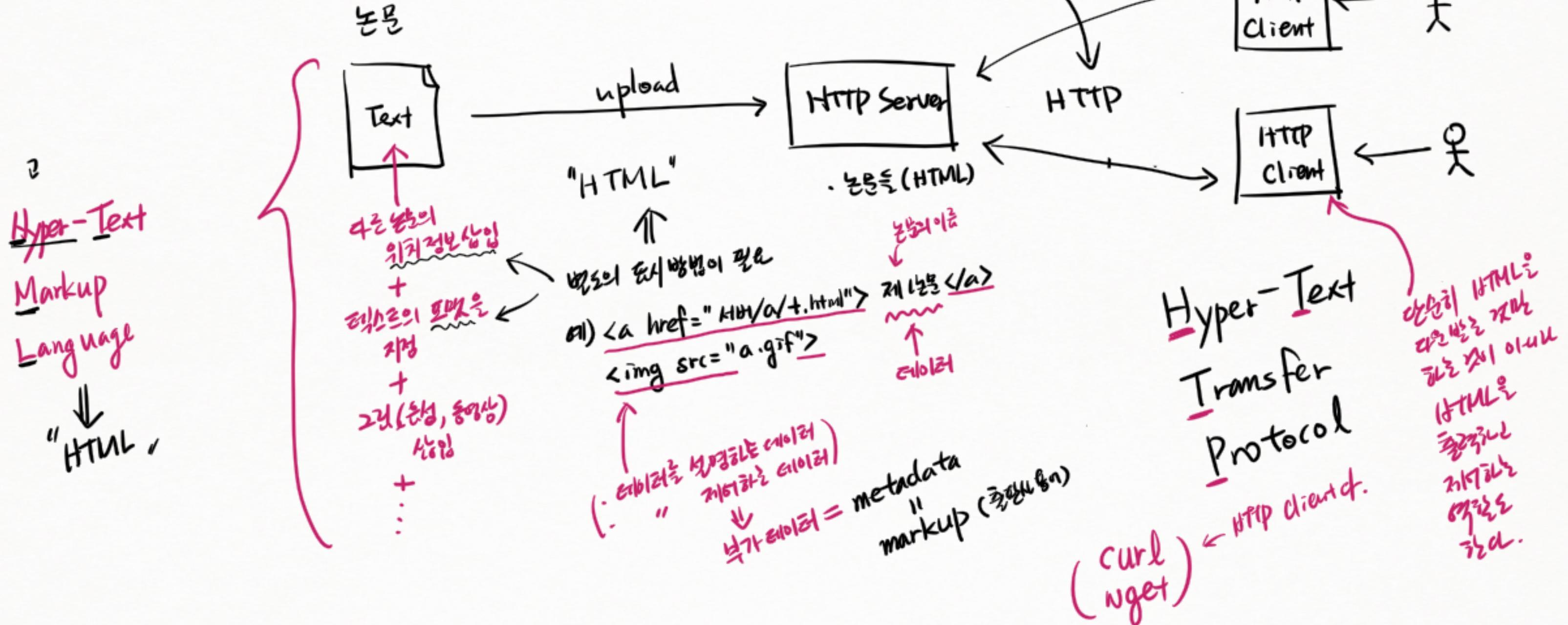
- 논문이 첨부하는 다른 논문을
다운로드하기 번거롭다
- 논문 안에 다른 논문이 업로드된
서버 주소가 많다.

통신 규칙이 아니라 Data를 송수신
"protocol"

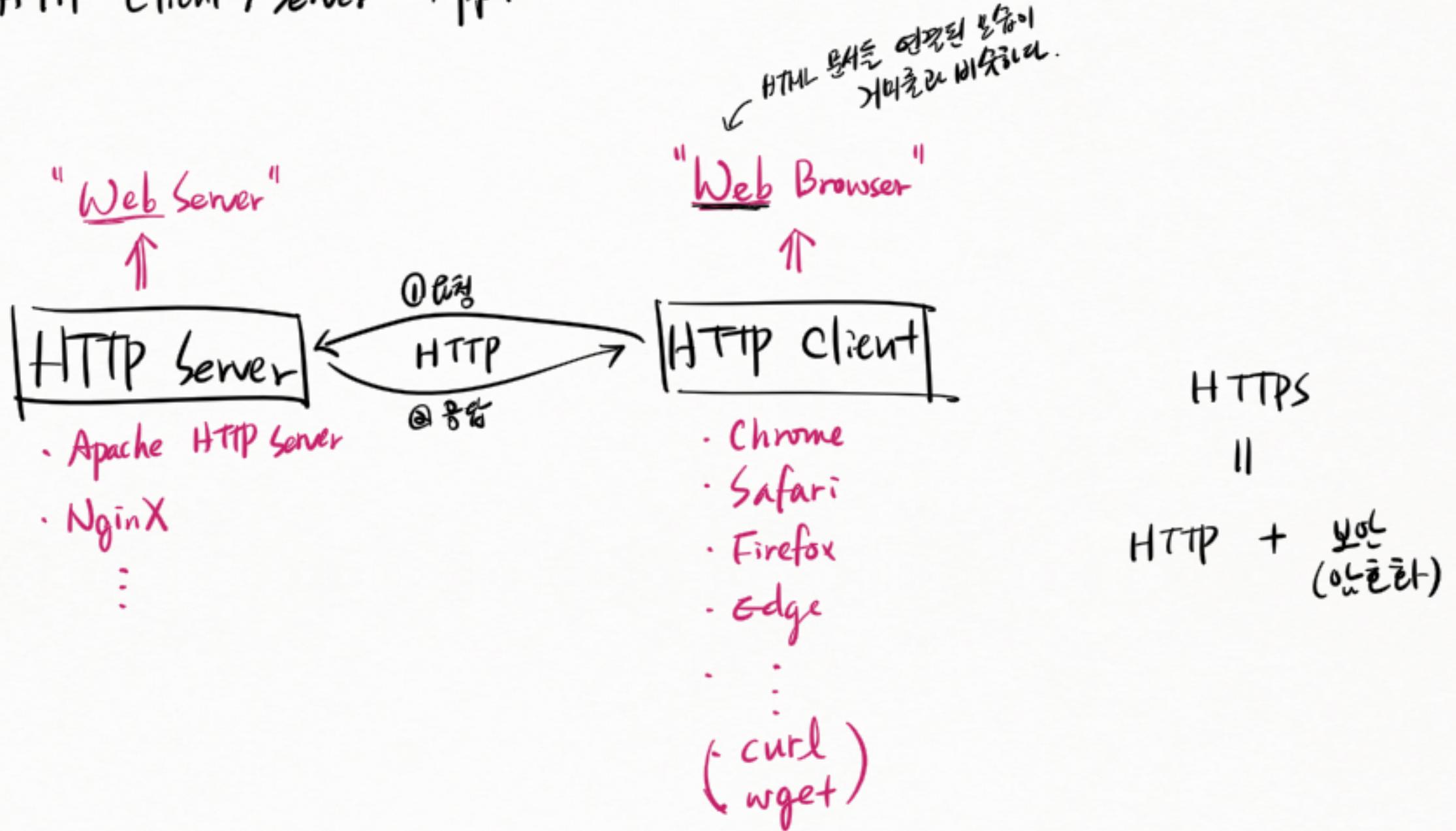
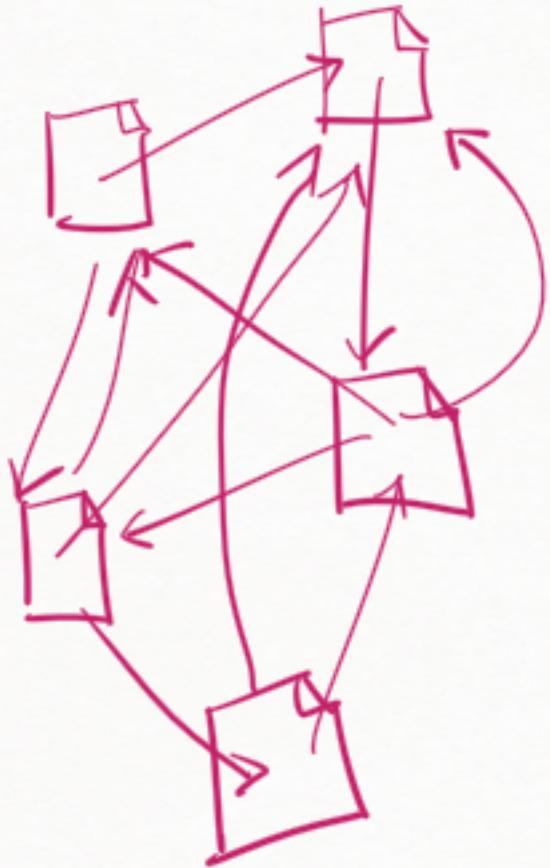
File Transfer Protocol

* HTML, HTTP

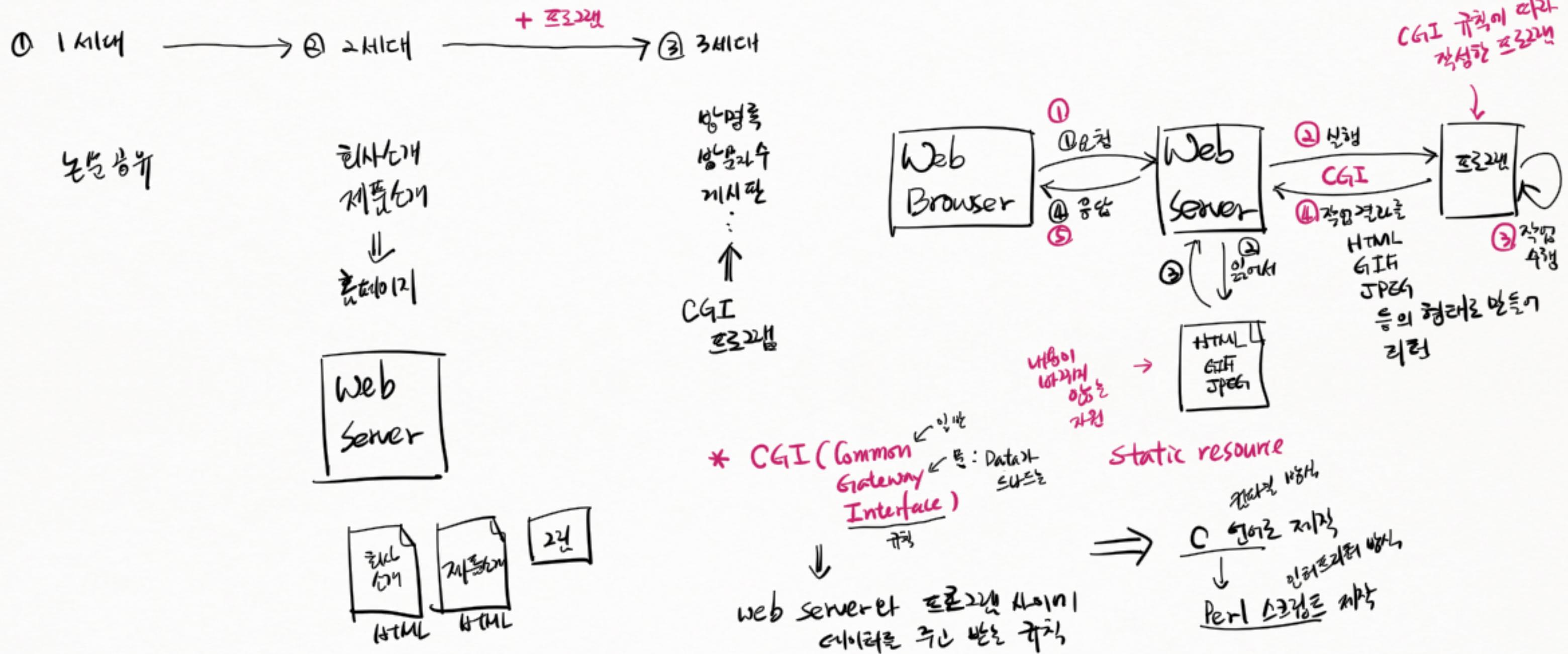
② HTTP, HTML 등장



* HTTP Client / Server App.



* Web 기초의 활용

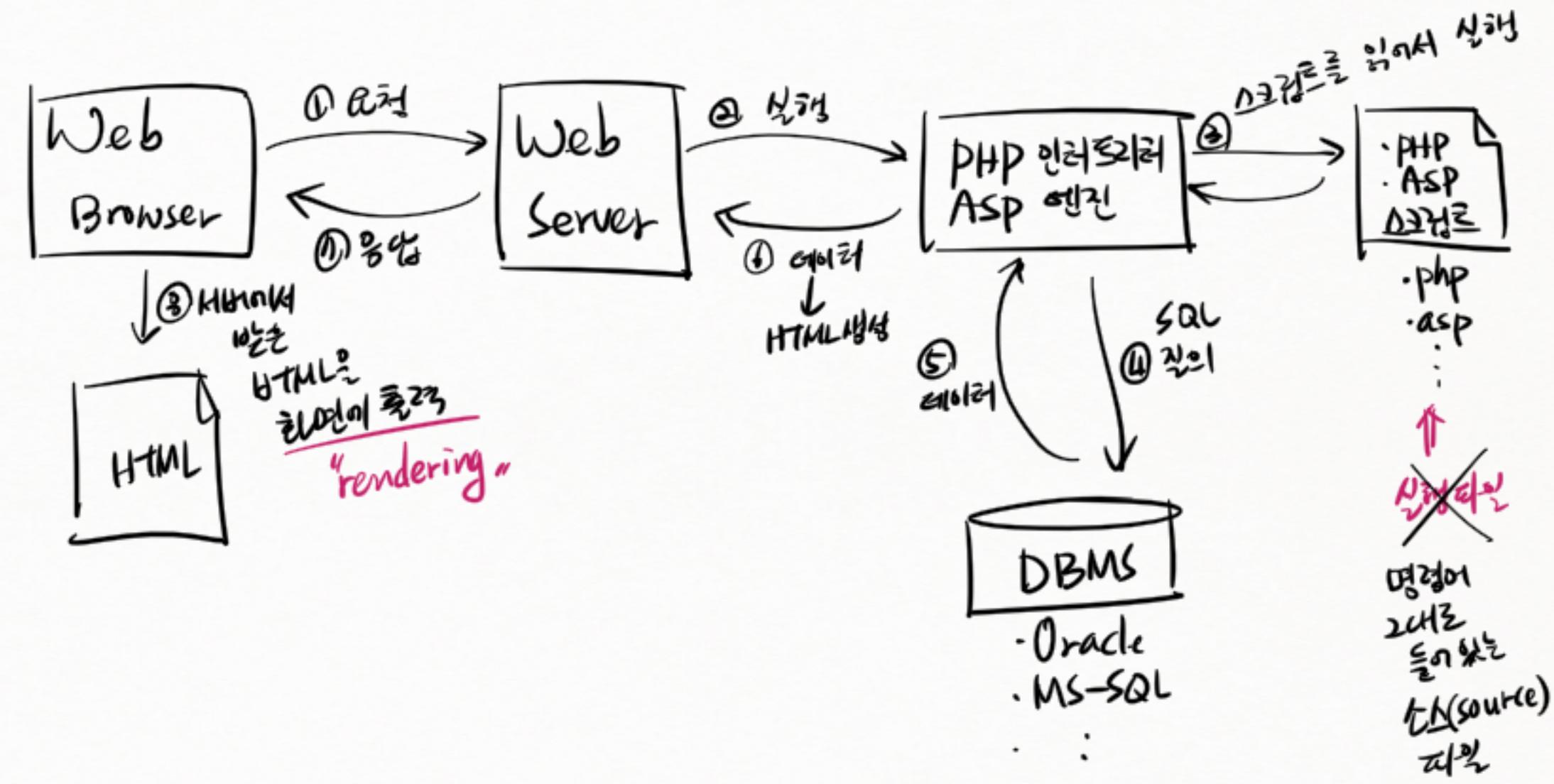


* Web 기술의 활용 II

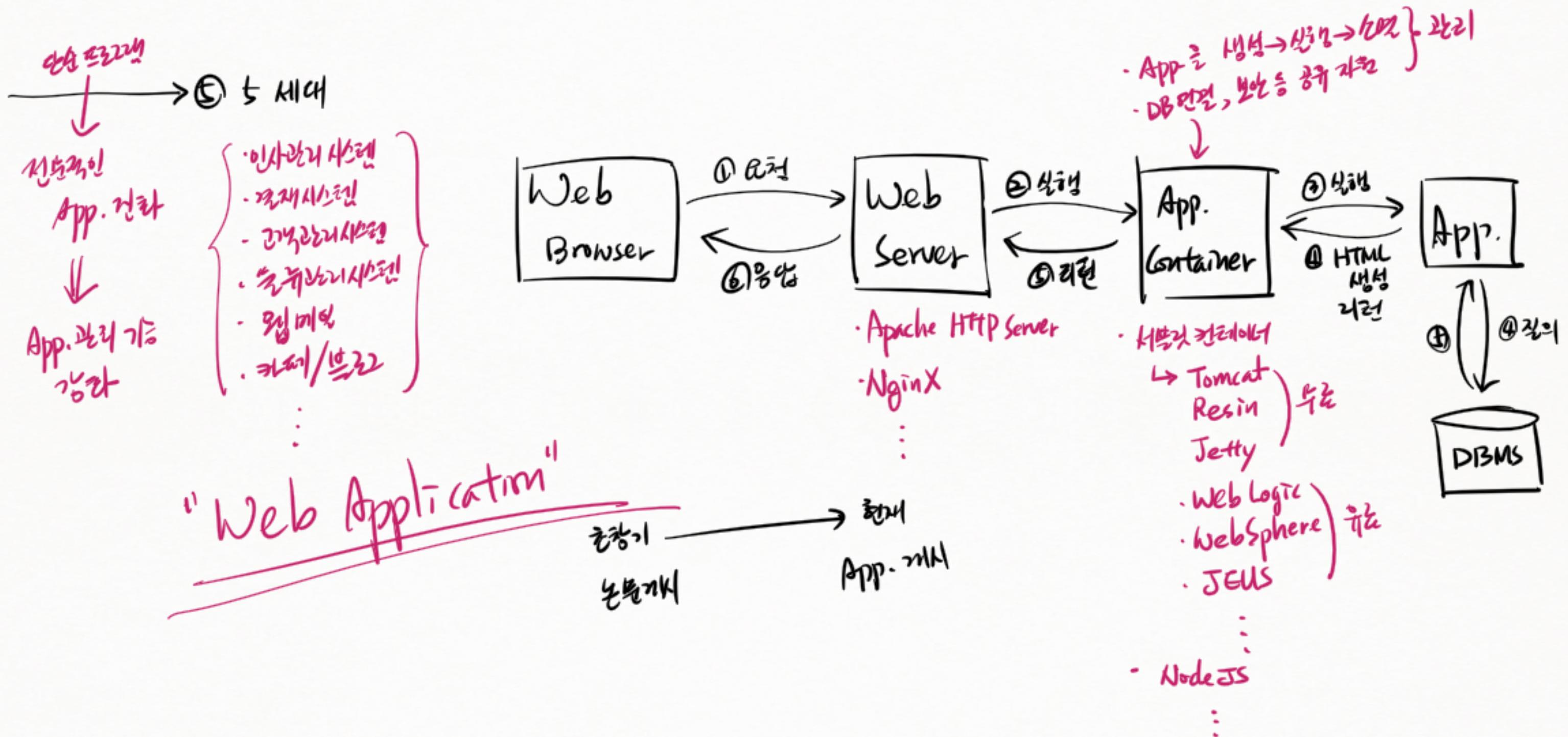
+ DBMS
+ 스크립트
→ ④ 웹서버

수정물
+
MS ASP 프로그래밍
PHP
:
+
DBMS

(01) 웹프로그래밍



* Web 기술의 활용 III



* Web Application 2가지 기술

[전면 만드는 기술]

CSS — 웹컨트의 디자인을
제작하는 기술

+

HTML ← 웹컨트를 구성하는
내용하는 기술

+

JavaScript ←
• 사용자의 입력에 응답
• 웹컨트를 제어

:

“Front-end 2가지”

← Full Stack
2가지

SI / SM

[서버를 만드는 기술]

✓ Java, JavaScript, PHP, ...

✓ SQL

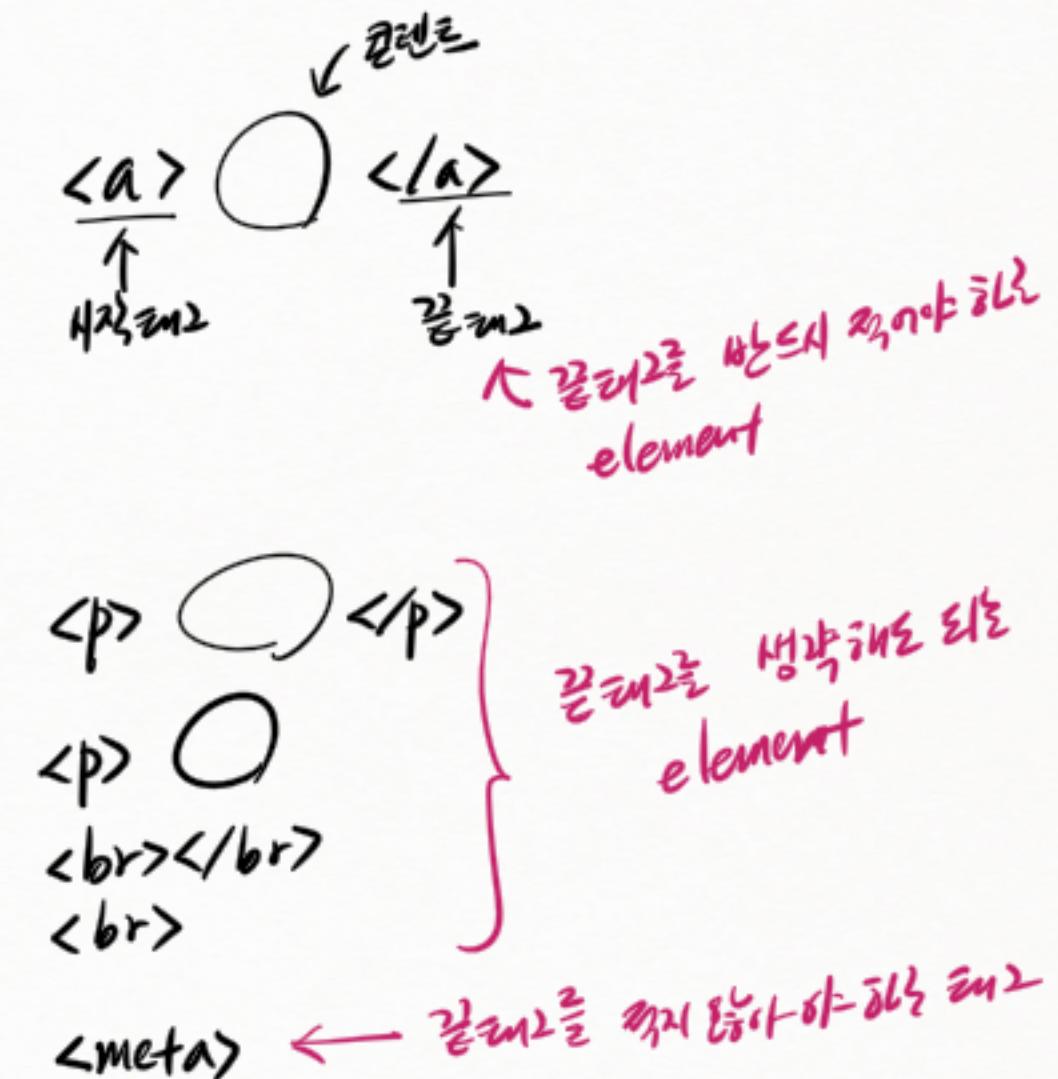
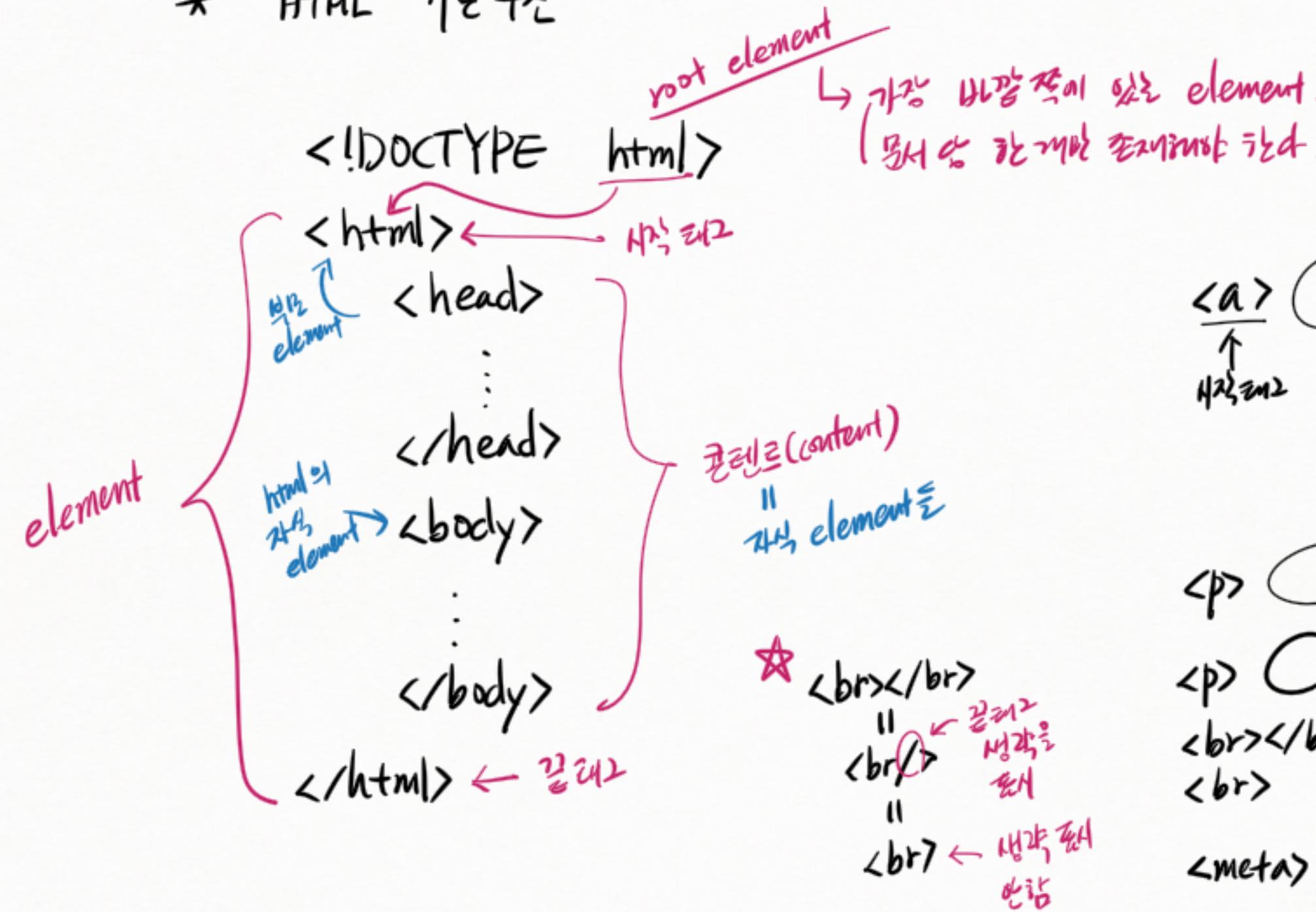
:



“Back-end 2가지”

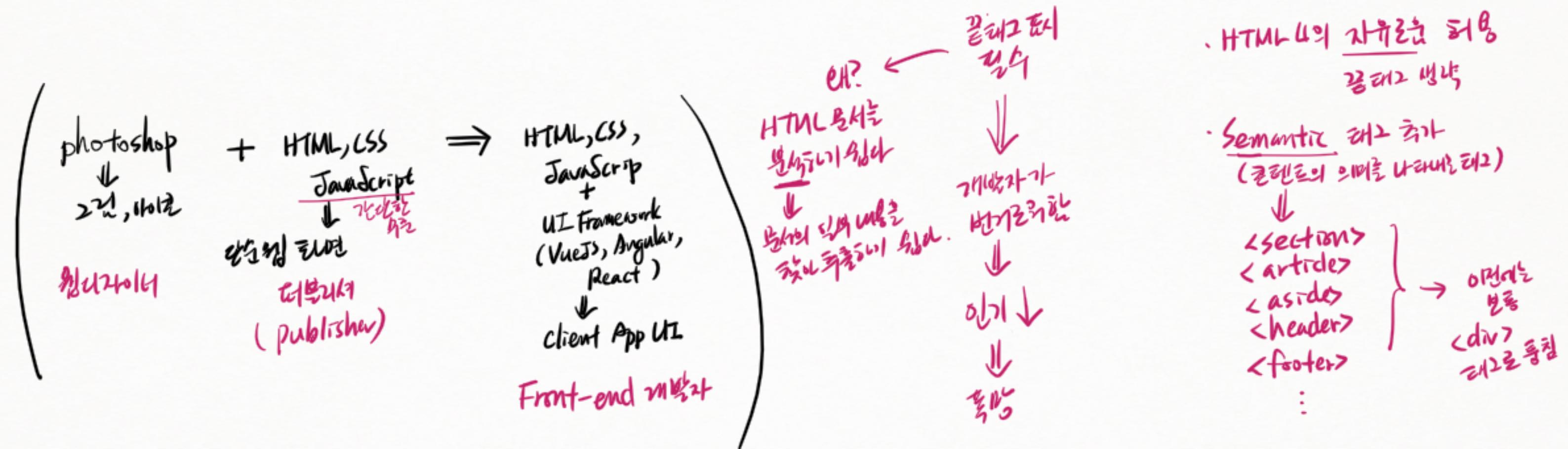
HTML

* HTML 기본 구조

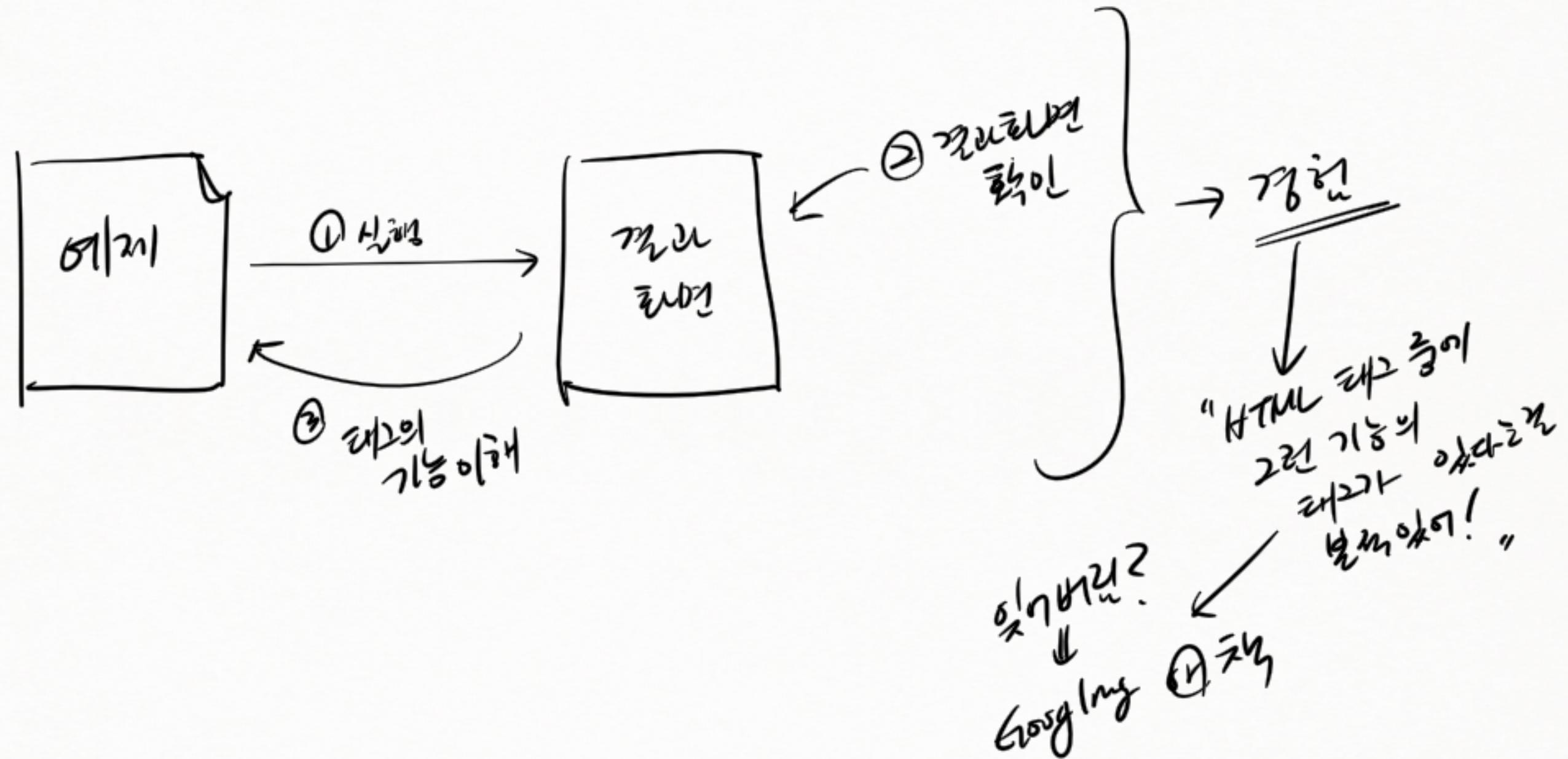


* HTML

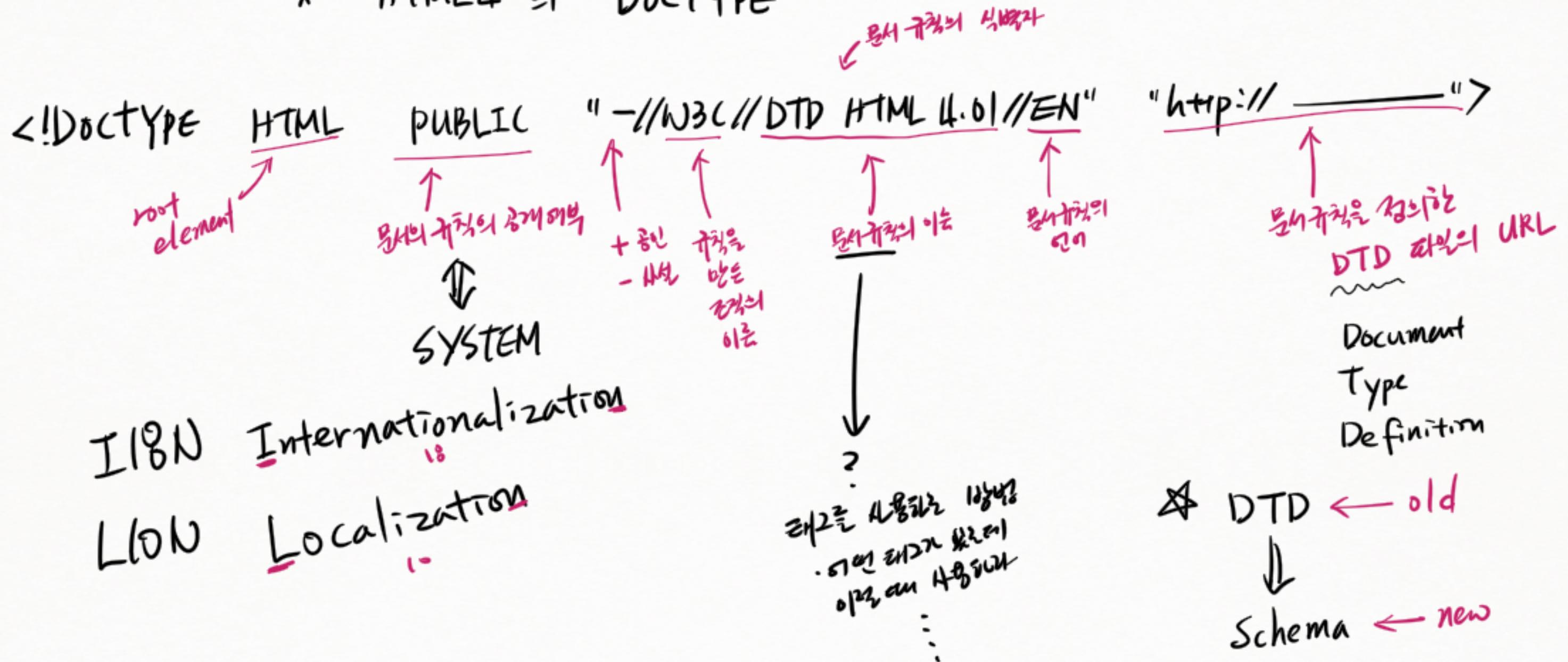
HTML 3.x → HTML 4.x → XHTML → HTML5



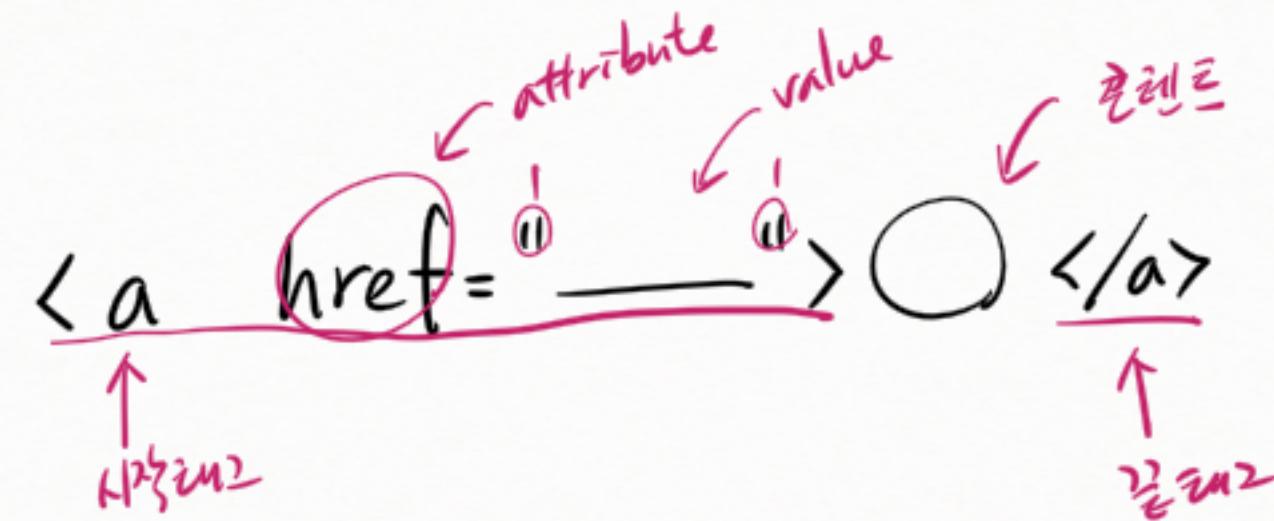
* HTML 풍부함



* HTML4 의 DOCTYPE



* tag et attribute



controls
selected
checked
readonly
:
} 키보드에
 마우스에
 터치 패널에
 현재 위치로
 기능을
 제공하는
 특수한 속성

* CSS

① <style> 태그

```
<style>
  =
</style>
```

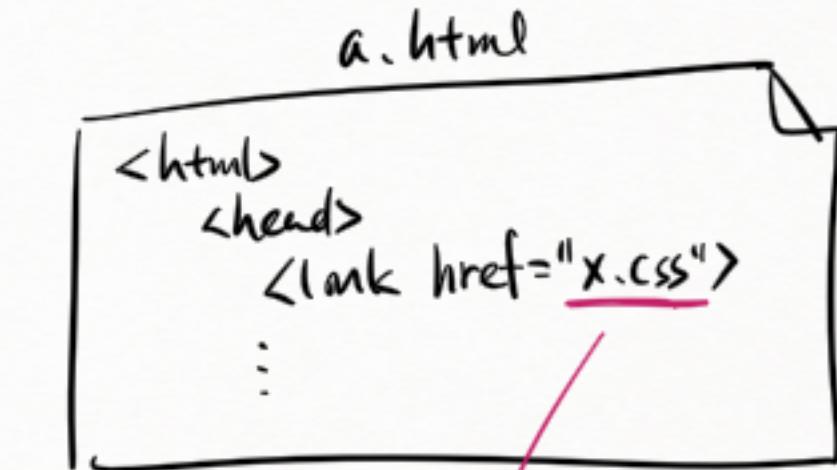
② inline 디자인

```
<tfoot style="—————" > 0 </tfoot>
```

inline style

이것!

③ 외부 css 적용



외부 HTML 파일에서
css를どのように 사용하는가?

x.css



* MIME Type

Multi-purpose
Internet
Mail
Extensions

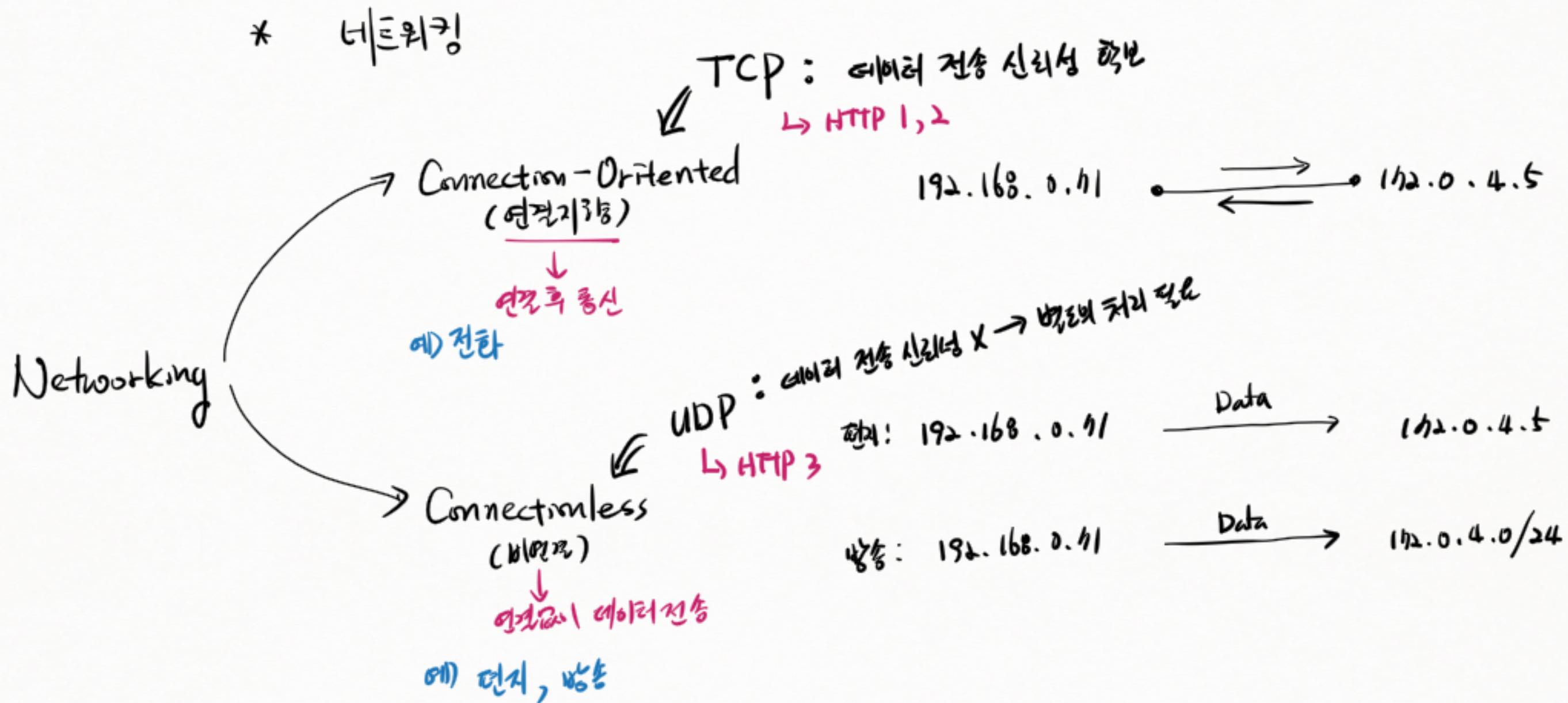
}
메일이 전부는 초 텐트가 어떤 형식인지
인터넷이나 메일과 같은 형식



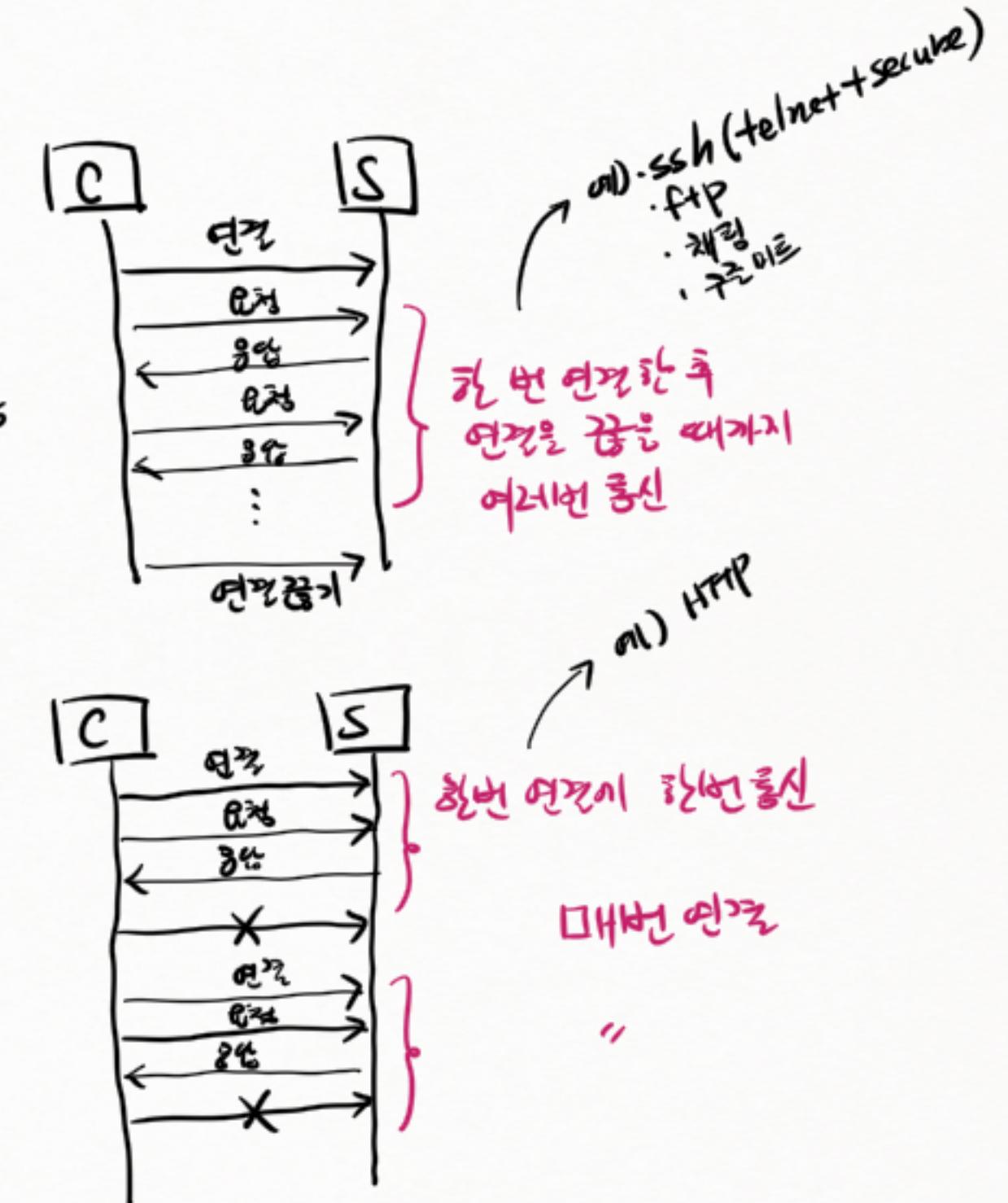
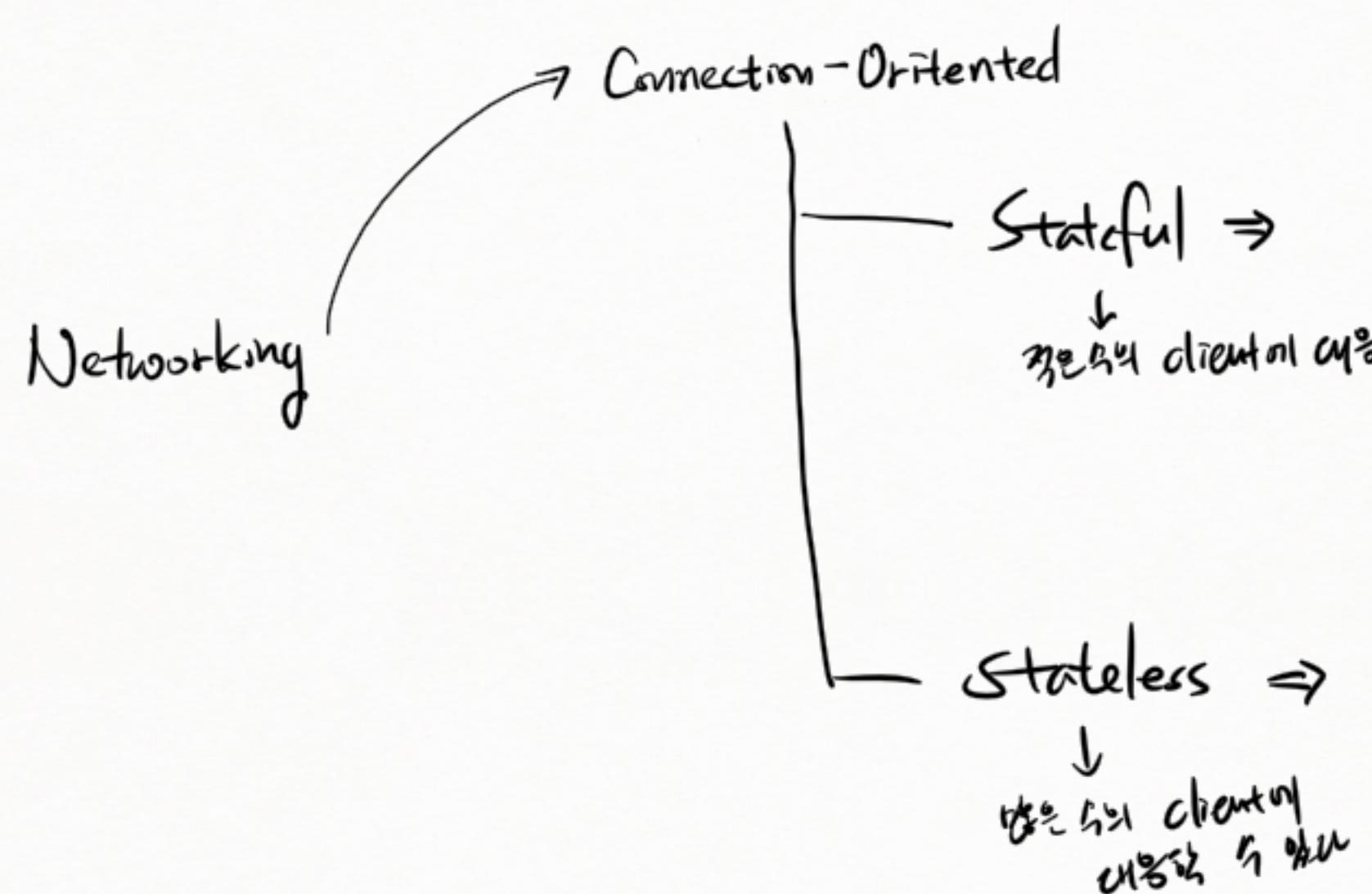
파일

파일 확장자는 Web 등 여러 가지 확장자로

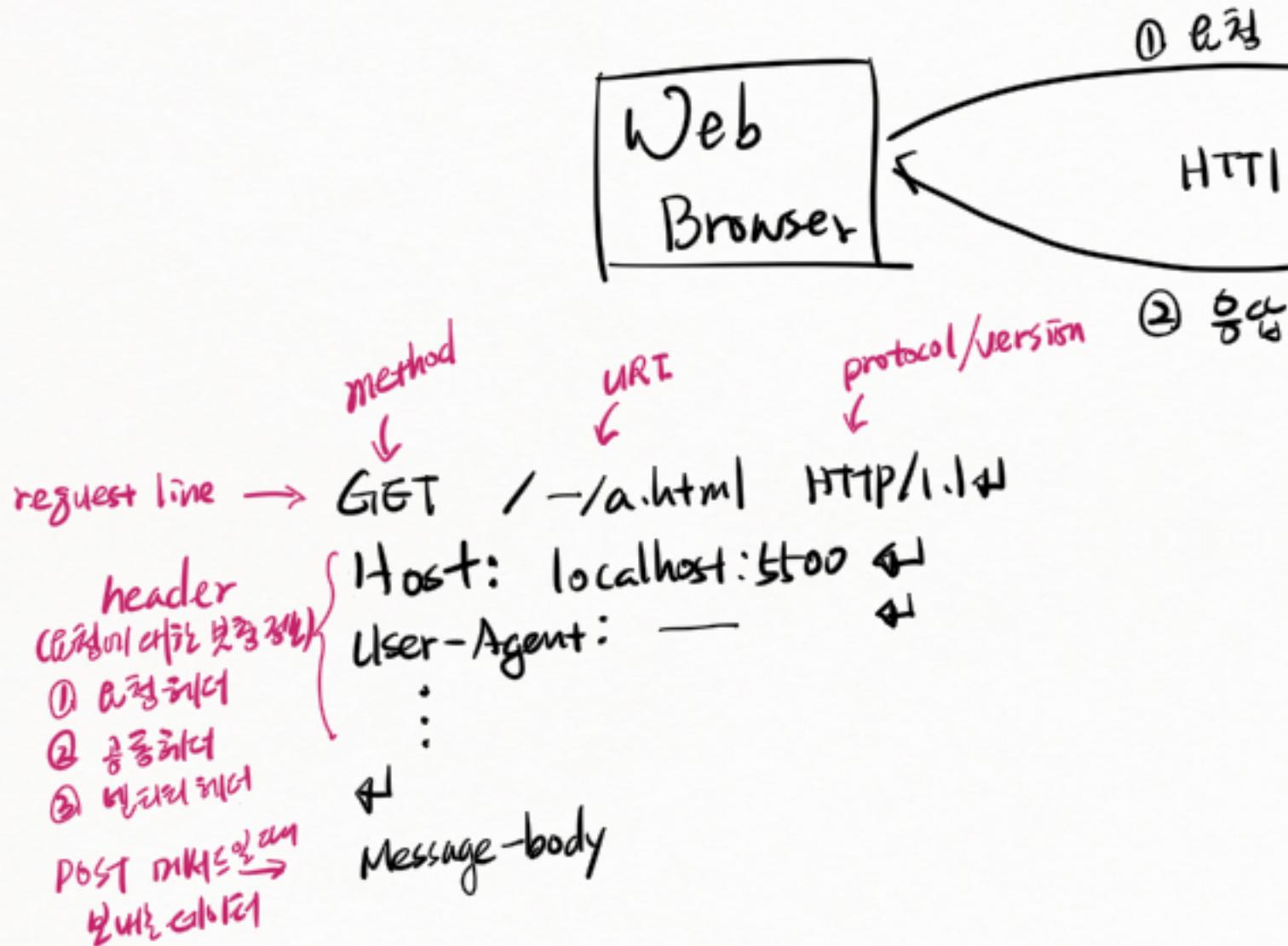
HTTP



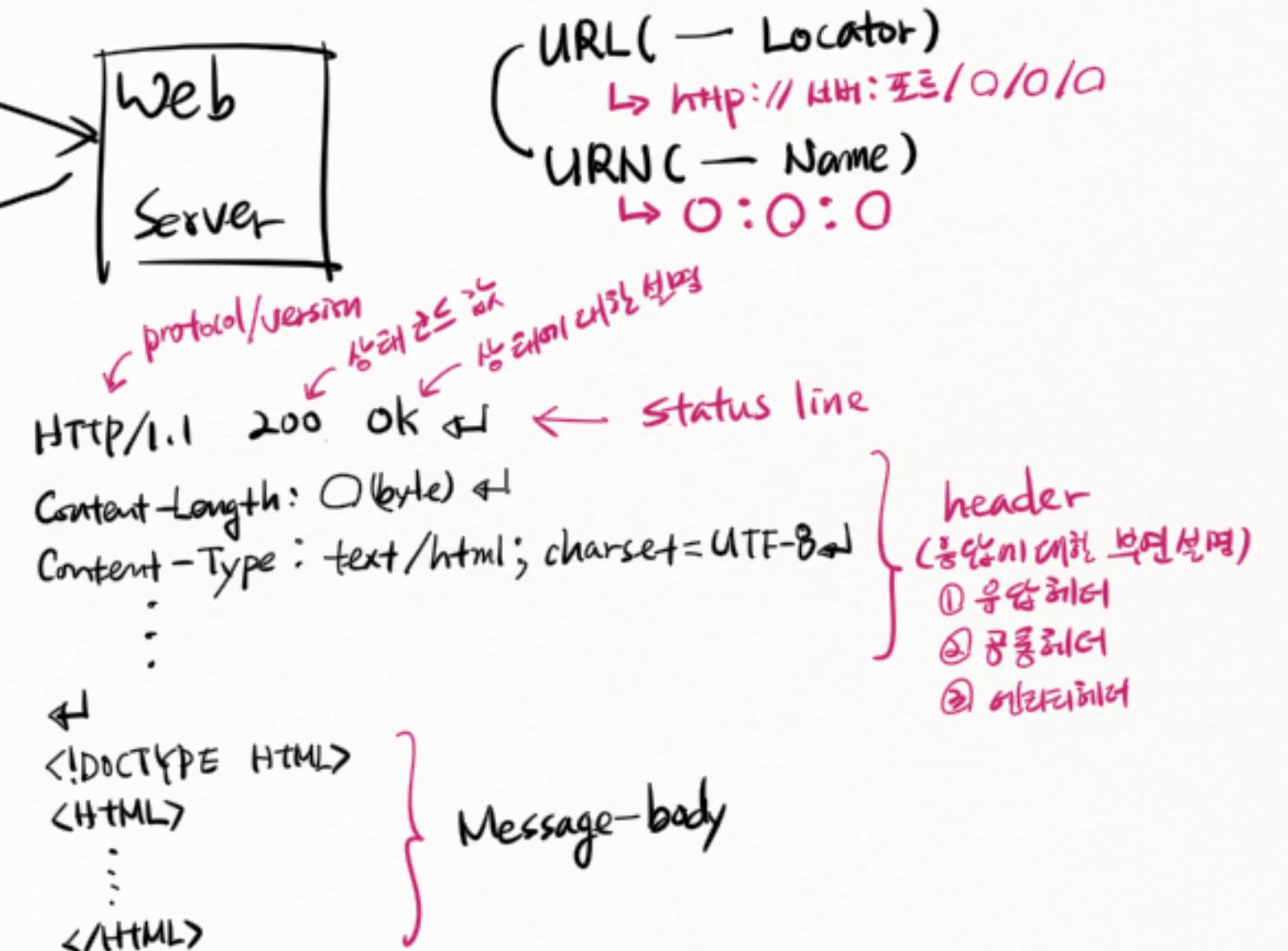
* 네트워킹



* HTTP 요청 / 응답

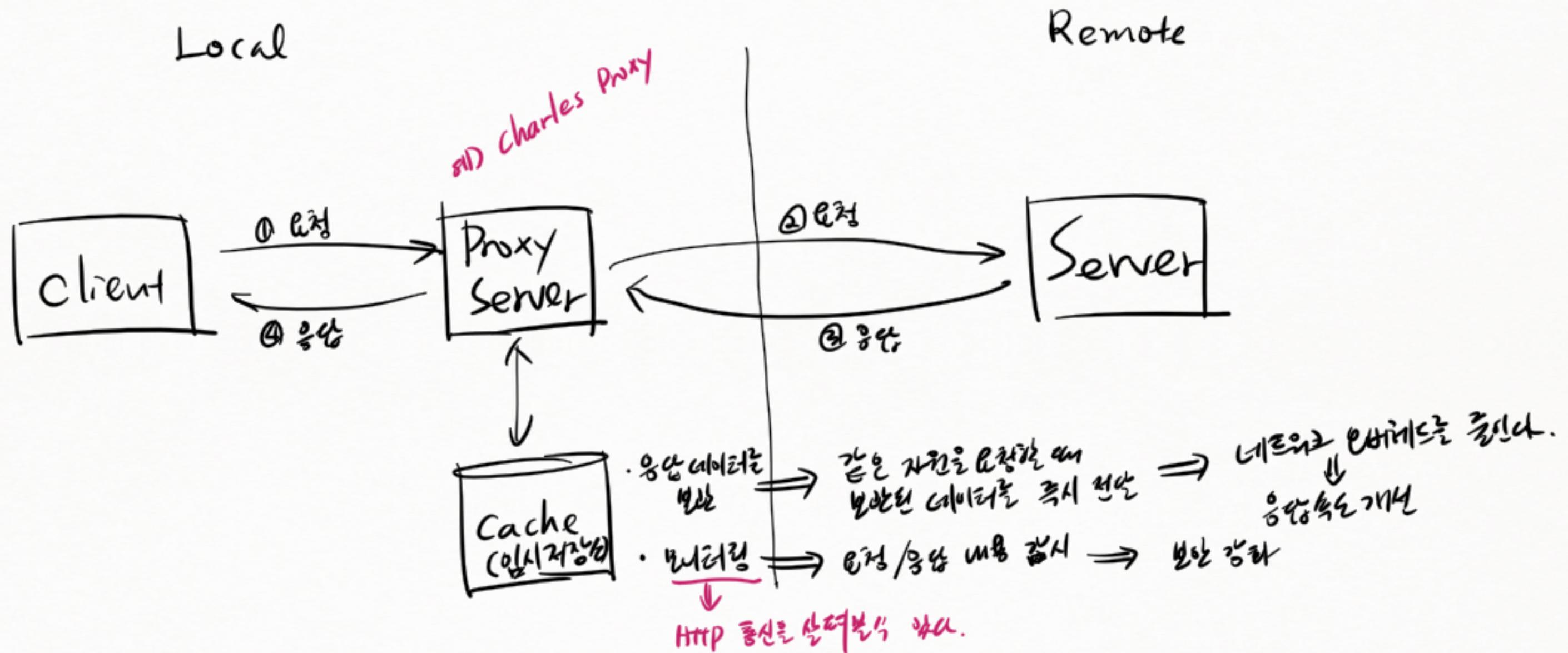


* URI (Uniform Resource Identifier)

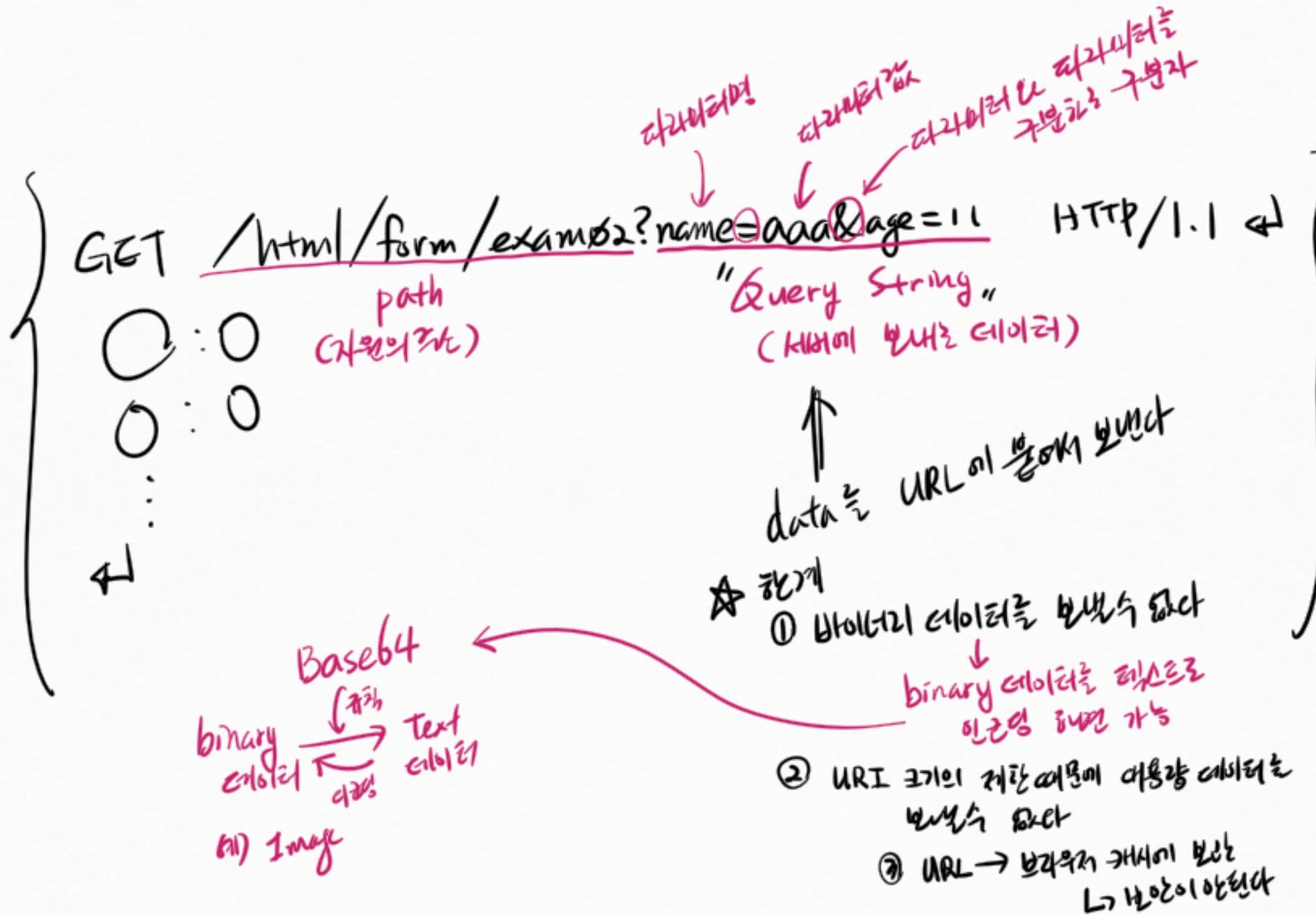


* Proxy HTTP

↑ client et Server 간에서 통신을 증가



* GET 요청



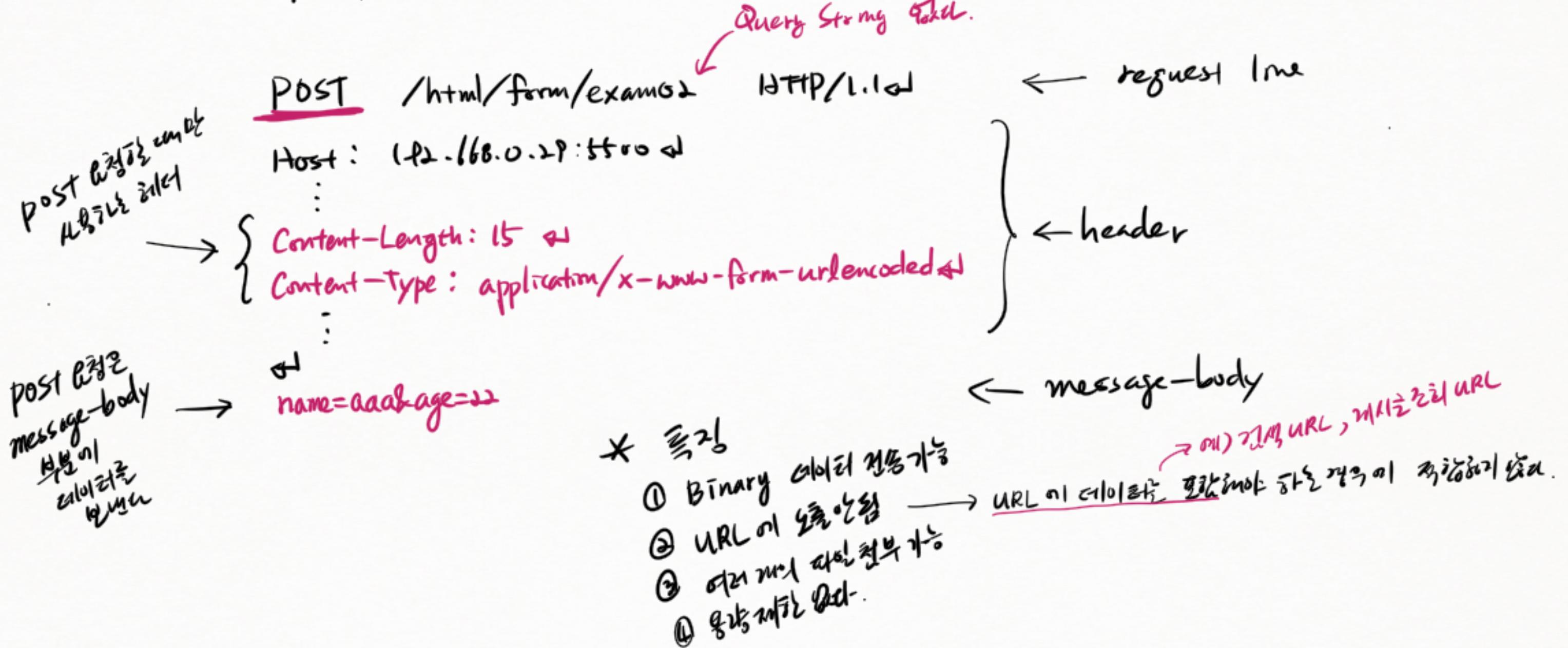
* text vs binary

- ① text
 - 일반 텍스트 파일 기호로 전송 가능할 때
 - 예) .txt, .rtf, .html, .css, .js, .xml, .json, .properties, .java, .py, ;

② binary

- byte 단위로 전송
- 일반 텍스트로 전송 불가
- 전용 APP. 사용
- 예) .jpg, .gif, .png, .mp3, .mp4, .avi, .doc

* POST 펴보기



* name 속성

```
<input type="text" name="title">
```



? title = ABC

자바스크립트

자바스크립트

* name 키워드 II

```
<input type="text" name="name">  
<input type="tel" name="phone">
```

abc

010-1111-2222



? name=abc & phone=010-1111-2222

css

* CSS : HTML element의 모양을 정의

↳ Cascading Style Sheet

selector

스마일 적용할 대상을
가리킨다.

.content header > span.title:hover {

background-color: red;

color: white;

}

↑
style name

↑ style value

pseudo selector
(선택자의 형태)

⇒ 셀렉터 { 스마일: 값; 스마일: 값; ... }

☆ ① selector 히어러

☆ ② specificity (스마일 적용순서)

③ box 다루는 방법
(레두리, 여백, 박스 레이아웃)

④ 폰트 다루는 방법

⑤ 색상 지정하는 방법

⑥ 배경 다루는 방법

⑦ block or inline 다루는 방법

⑧ 위치 조정하는 방법



* Selector

* {} — }

* {} — }

태그명 {} — }

body {} — }

태그명, 태그명, 태그명 {} — }

img, ul {} — }

.2중명 {} — }
class

.c1, .c2 {} — }

#태그아이디 {} — }

#content {} — }

-[속성명=값] {} — }

-:상태명 {} — }
pseudo selector

li:hover {} — }

진상 자손 {} — }

#content li {} — }

부모 > 자식 {} — }

ul > li {} — }

태그 + 다음태그 {} — }

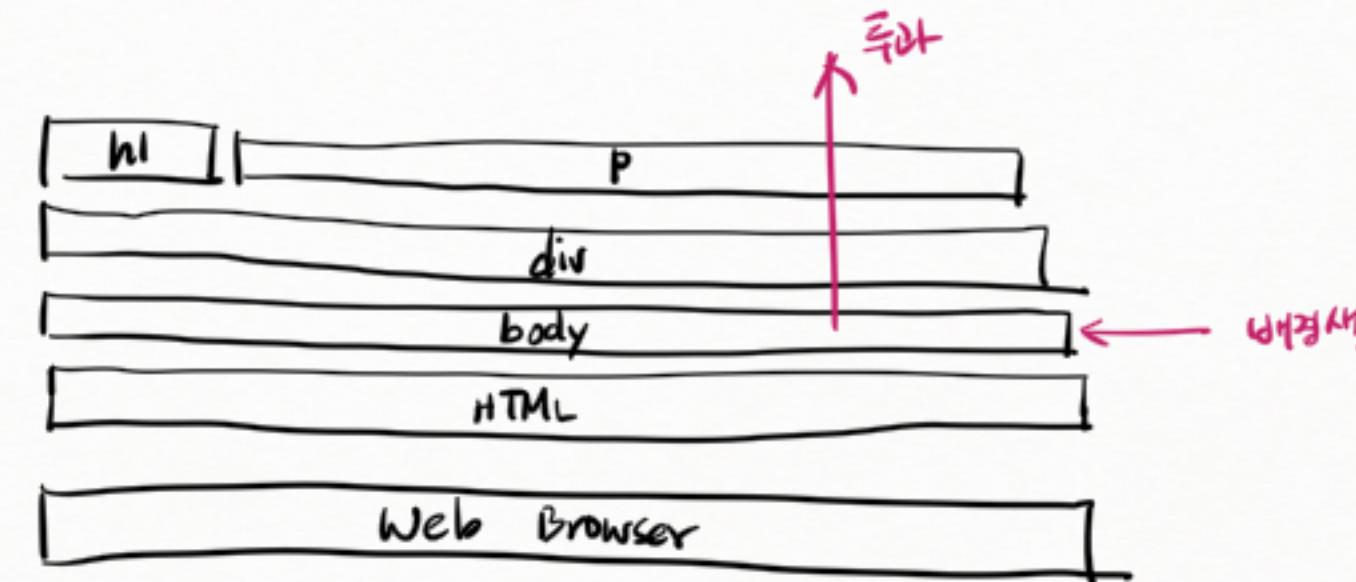
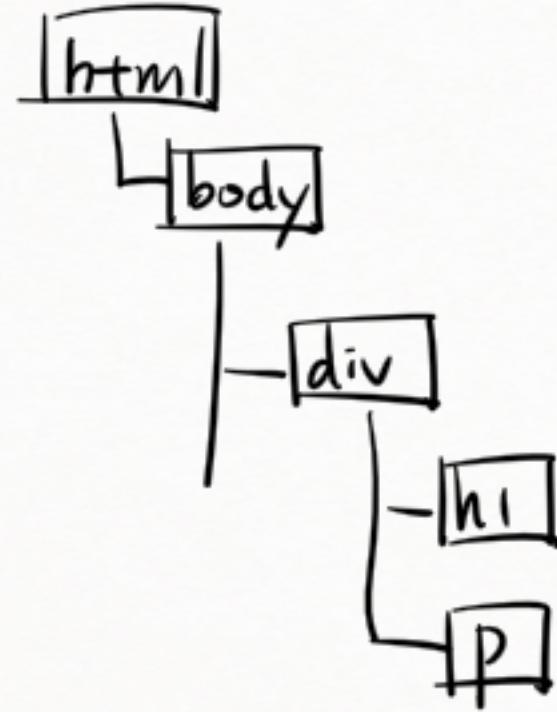
img + ul {} — }

대상자 조건 값 ... {} — }

div#menu li:c2 {} — }

input[type="text"] {} — }

* 부록 태그와 자식 태그의 위치 찾기



* CSS specificity : 스타일 적용 순서 낮 → 높

* → 0

id, pseudo selector → 1

class, tag → 10

attribute → 100

inline style → 1000

* {}

h1 {} a:hover {}

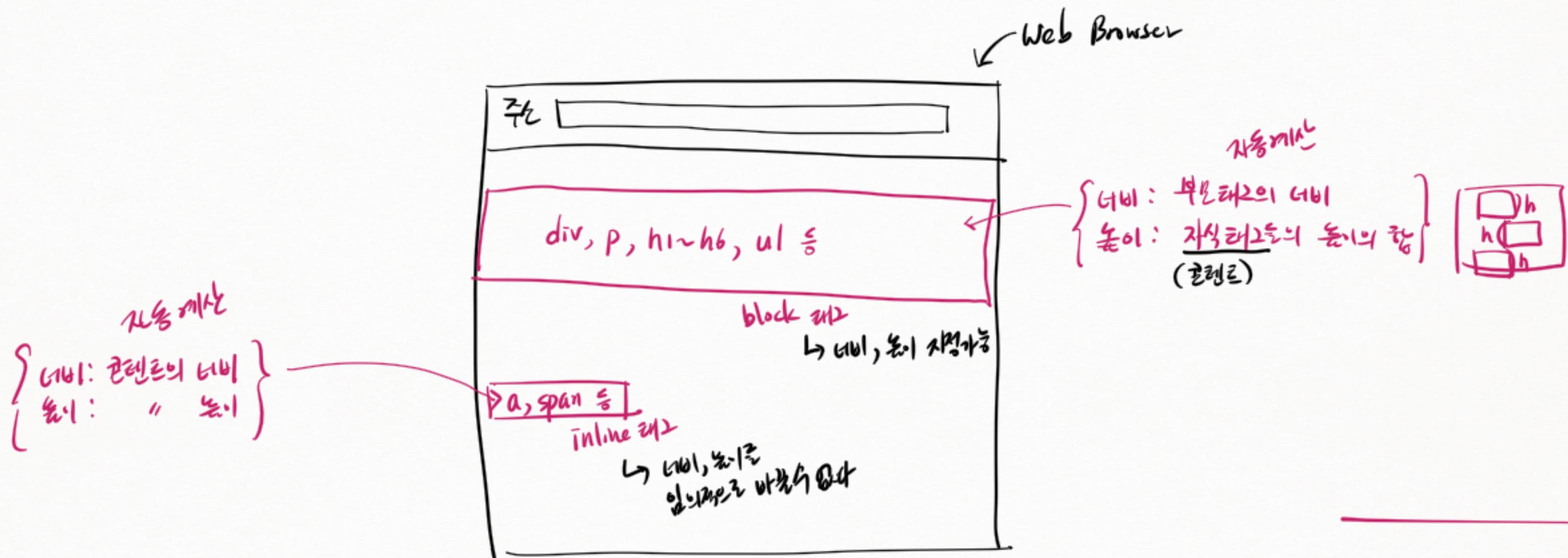
.cl {} input[readonly] {}

#menu {}

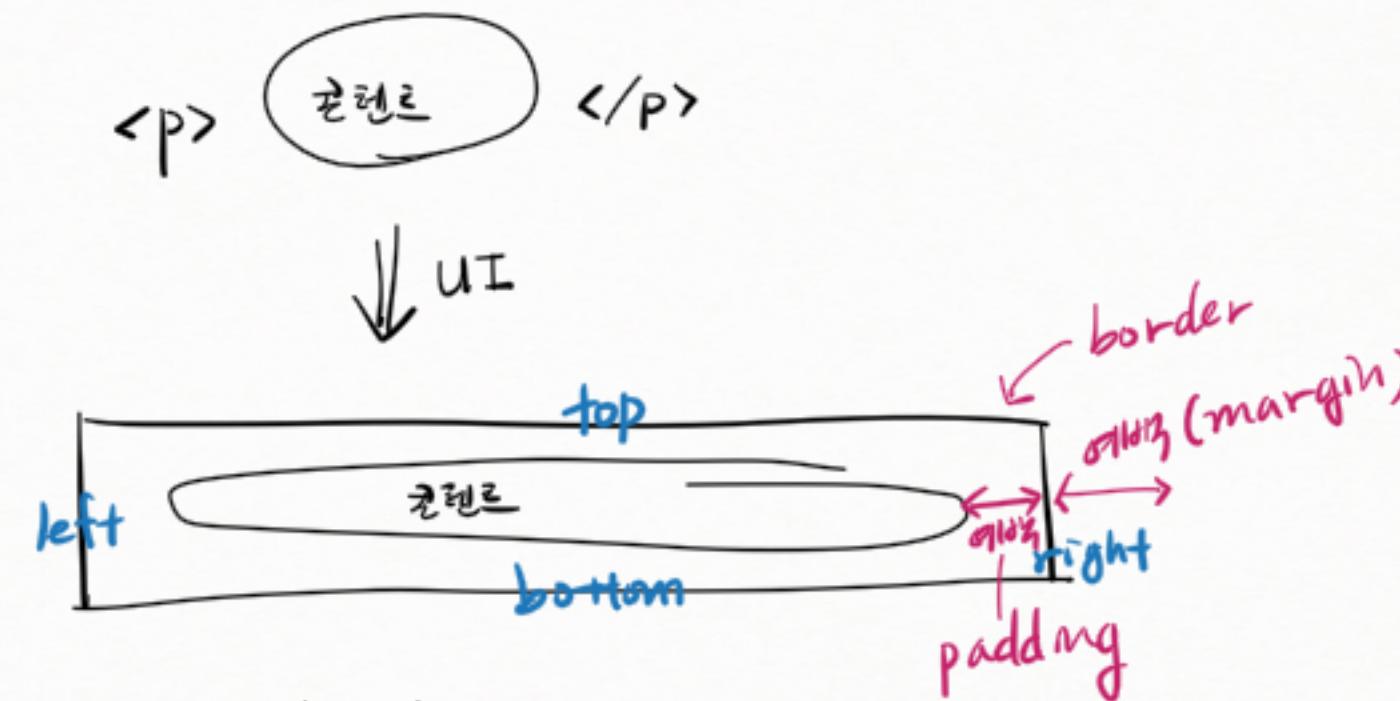
 ()

인라인 스타일

* 태그의 너비, 높이



* 레이아웃 (border)



right
bottom color dotted
border-top-style: solid;
left width dashed
:

짧은 명령
border-top: color style width;
border-bottom:
border-right:
border-left:

border-style: solid;
모든 방향의 레이아웃 스타일

더 짧은 명령
border: color style width;

레이아웃 방향

* Raster 폰트 Vector 폰트

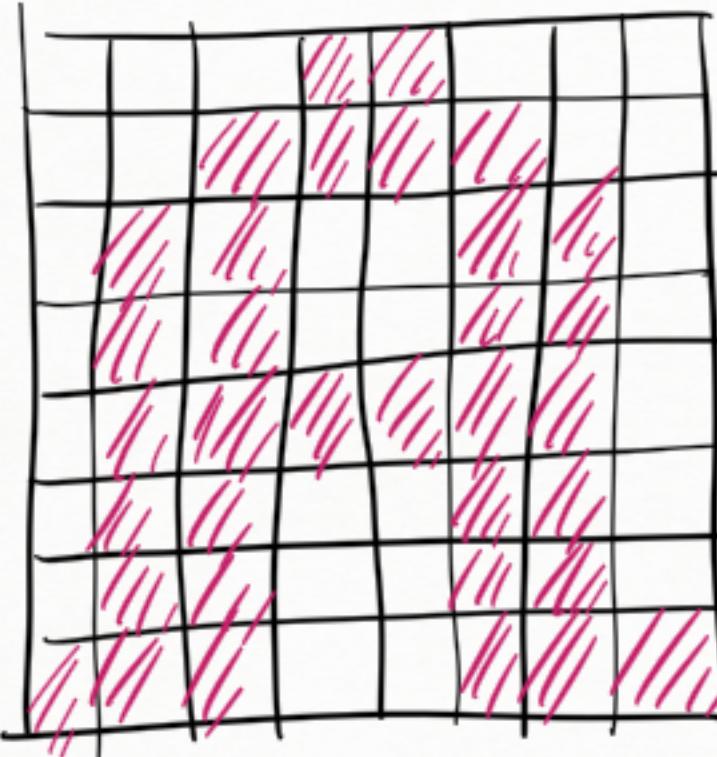
↳ 비트맵 이미지 예) .bmp, .gif, .png, .jpg ...

↳ 벡터이미지 예) 층집마크, 캐드

Raster 폰트

↳ 퍼센트 단위로 글자를 만든다

- 한글 속도가 빠르다
- 폰트크기에서 따라
각 문자를 만든다
- 정해진 크기보다
더 크게 한글 속도를
얻을 때 각 문자의
크기를 늘리기 때문이다
제작 현상 발생
- 이미지의 불확정화
상관없이
파일크기는 같다.



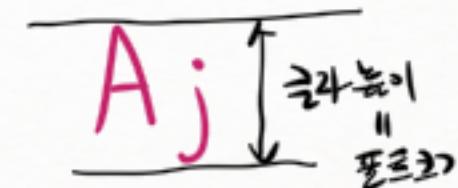
(1) COURIER

Vector 폰트

- 글자를 그리는 명령어를 작성한다
- 글자를 그리는 명령을 수행

* 폰트크기 = 높이

4~5 line
3~6 line
(3,3) (1,8) line
⋮



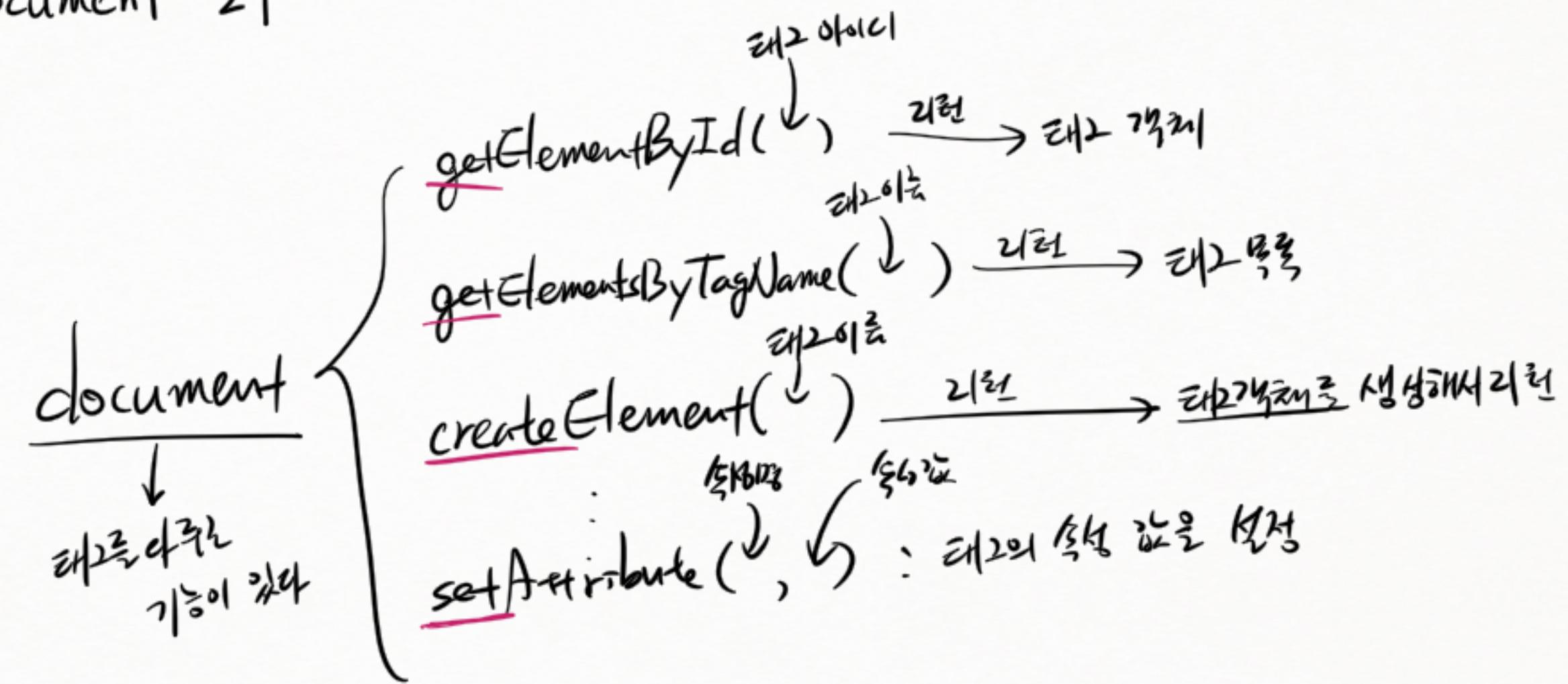
(1) True-Type 폰트
courier new

한글 속도가 느린다
(같은 CPU가 빠르기 때문에
상관하지 않는다)

- 글자 크기를 늘리더라도
명령을 통해 그리기 때문에
제작 현상이 발생하지
않는다.

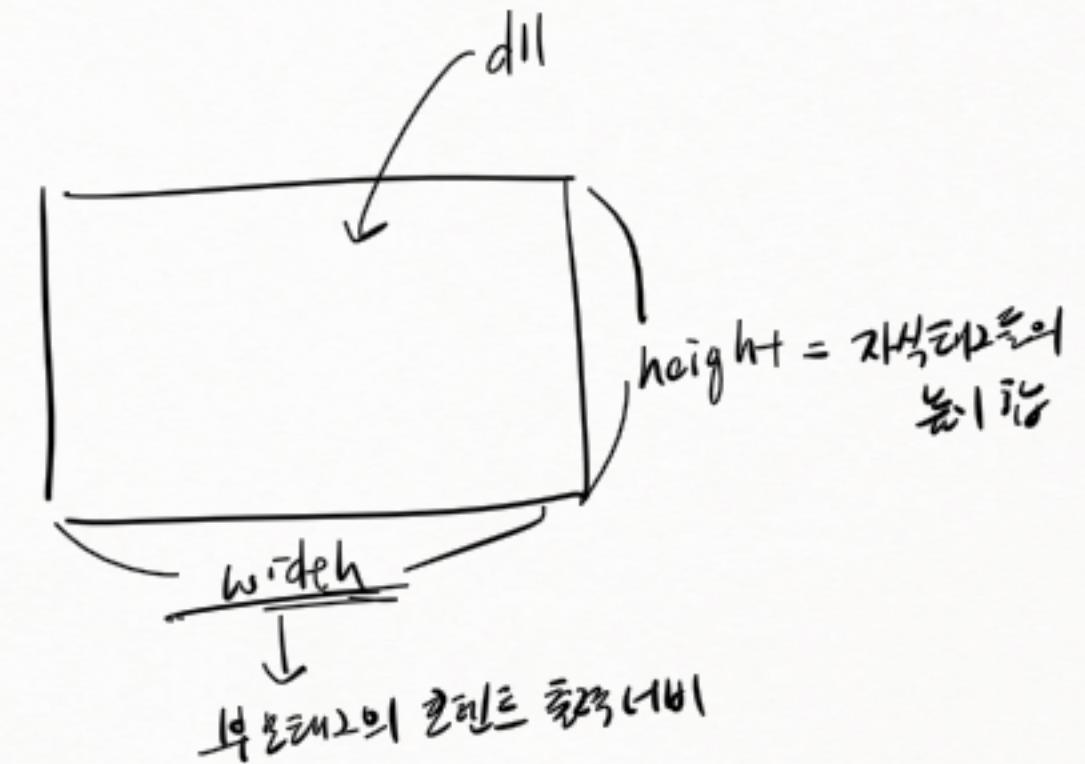
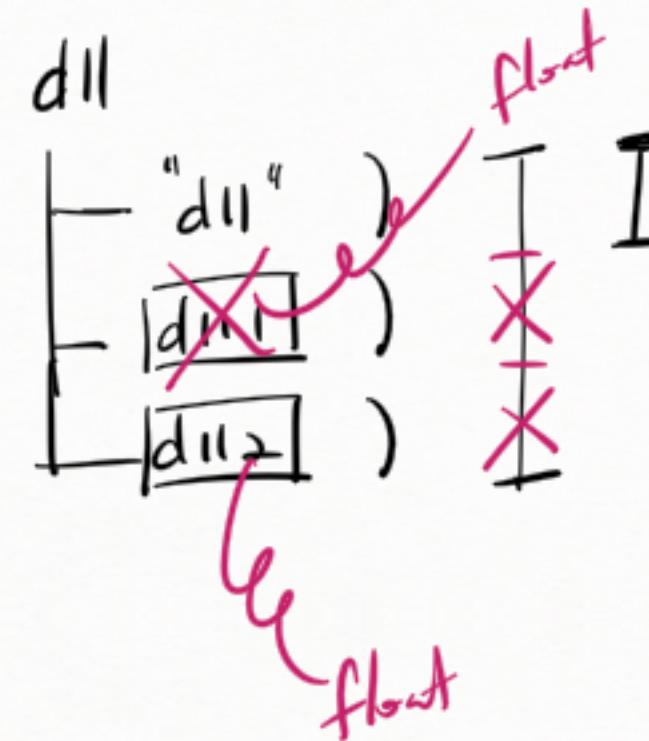
- 이미지 크기 → 파일크기 ↓
" 불확정화 → " ↑

* document 속성



* float 와 absolute

↳ 부모태그의 크기를 계산할 때 float는 absolute 태그를 제외한다.



* CSS와 JavaScript 코드의 분리

