web client 기술 = 웹서버에서 해당 파일 찾아서 브라우저 전달 - 웹브라우저 실행

html5= html 요소 + css 요소 + java script

java script = 부품의 동작 지정 역할 *.js

jquery – dom과 이벤트 처리 등을 쉽게 하도록 도와주는 자바 스크립트 함수 라이브러리
1> 쉬운 문서 객체 모델(dom)
2> 쉬운 이벤트 처리
3> 쉬운 css
4> 쉬운 ajax 처리

-jquery 실행 가능한 html 파일

```
<script src=js/jquery-3.6.0.min.js ></script>
<script>
$(document).ready(function(){
//////////////
});
</script>

===>
jQuery(document).ready()...
```

- JQUERY 문법

```
$( selector).함수(매개변수지정);
```

1> jquery selector

2> 함수

2-1. 1개 함수 setter/getter 동시 사용

css:스타일 지정	$("").css("css이름", "값"); -->setter 용도 $("h1").css("color", function(index) { ..... return 색상이름 }); -->setter 용도 $("").css({css이름동일변수:"값" , , , }); -->setter 용도	$("").css("css이름"); -->getter 용도 $("h1").css("color"); -->getter 용도+반복문(또는 each함수) $("h1").each(function(index){ });
스타일지정	<style> .red {color:red;} $('H1').addClass("RED"); removeClass("red") toggleClass("red"d) <h1 > aaa</H1>	
HTML태그의 속성:attr() removeAttr()	attr("속성이름", "값")	attr("속성이름") removeAttr("속성이름")

텍스트 제어
text()
html()
append()
	$("div").text("setter- html해석x")
	$("div").html("setter-html해석o")
	$('div').innerHTML = "오류발생";
	
	document.querySelector("div").innerHTML = " 제목 ";
		alert($("div").text());
		alert($("div").html());
		alert(document.querySelector("div").innerHTML)
		  .....

	document.querySelector("div").text Content = " 제목 ";	
이벤트처리	document.querySelector ("input[type=button]") .addEeventListener ('click', 함수, false); document.querySelector ("input[type=button]") .removeEventListener ('click', 함수, false);	$("input:button").on ('click', 함수(){}); $("input:button").off ('click'); $("input:button").click (함수(){}); 385p 표11-4
반복	each(function() {...})	
시각효과	$().hide(); $().fadeOut(1/1000초); $().slideDown() $().animate(function() { 정의 })	$().show() $().fadeIn() $().slideUp()

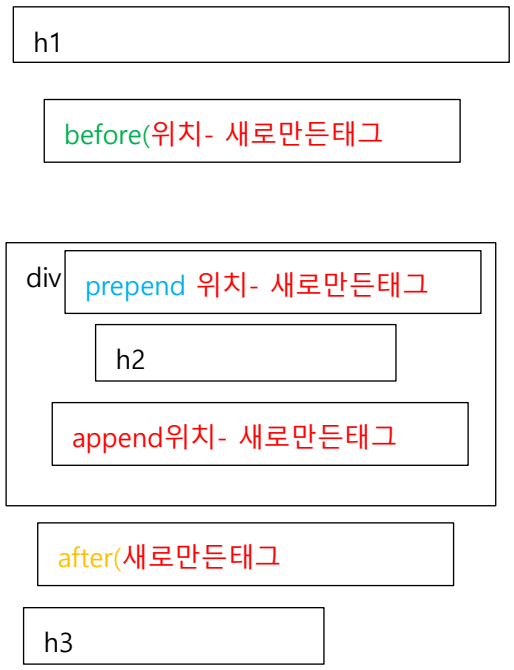
<table>
<tr>
<td>

<h1>
** <h2>...</h2> **
<h3>

$('div').append(새로만든태그)
$('div').prepend(새로만든태그)
$('div').before(새로만든태그)
$('div').after(새로만든태그)

$(새로만든태그).appendTo('div')

</td>
<td>



</td>
</tr>
</table>

특정 태그 내부에 다른 태그 포함 --> $("부모").append("자식") text/html/append

423p

무한 스크롤

viewport – url, 타이틀탭 제외 브라우저 왼쪽상단

scroll – 위 가려진 부분 높이

html파일 컨텐츠 높이

html파일 컨텐츠 높이 = 브라우저 높이==> 스크롤바 없다

html파일 컨텐츠 높이 > 브라우저 높이==> 스크롤바 있다

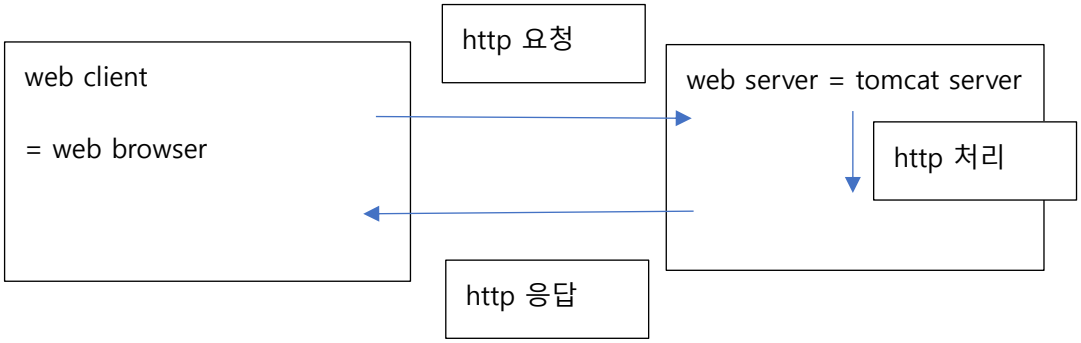
위 가려진 부분 + 브라우저 높이 == html파일 컨텐츠 높이==> 스크롤바 무한 생성

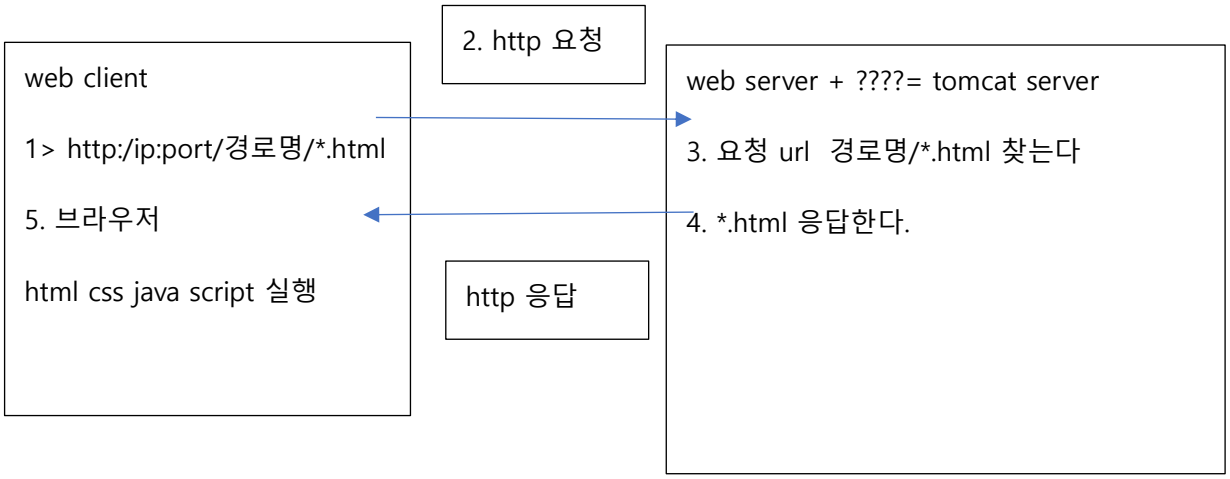
html + css + java script + jquery

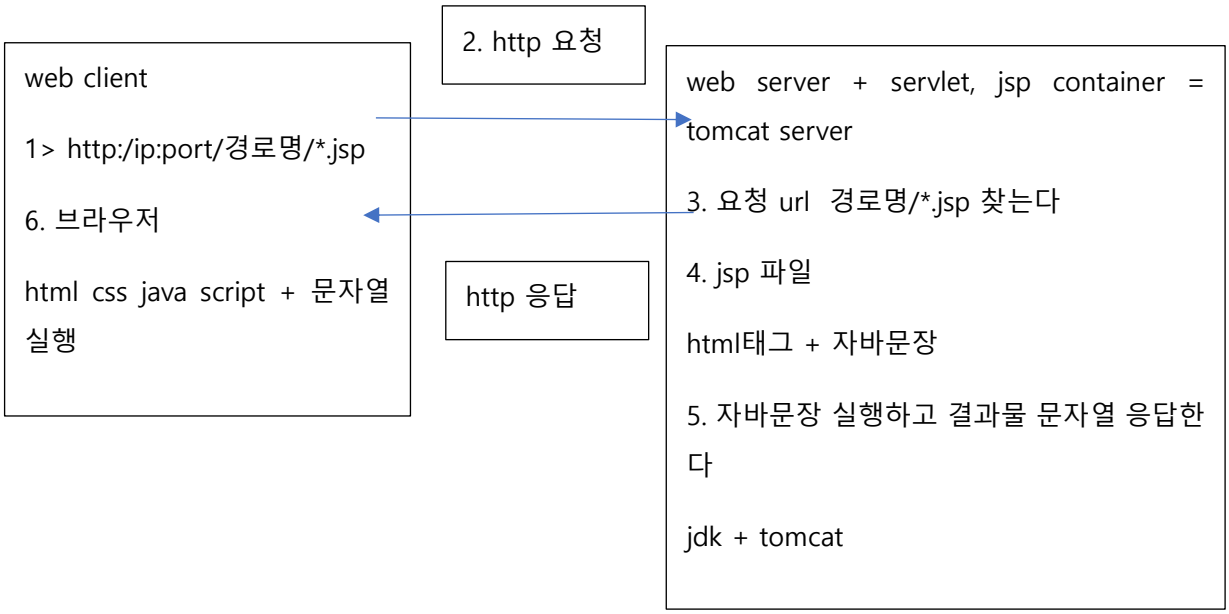
java / sql / html / css / java script

web server

web server = java 구현 기술 = servlet, jsp, spring(di, mvc , 다른 기술 연동), mybatis







1-4장 빠르게 이해

1장 2 3장

4장

1>이클립스 dynamic web project

프로젝트이름 = context이름 = web application 이름	jre system library-> jdk 라이브러리(모든 자바 파일 라이브러리) tomcat library->(servlet, jsp 라이브러리)
	src/main/java₩패키지명₩*.java (DTO, DAO, SERVLET....)
	src/main/webapp/*.html/*.css/*.js/*.jpg/*.mp3 src/main/webapp/WEB-INF/lib src/main/webapp/WEB-INF/web.xml

http://localhost:8080/servlettest/FirstServlet

http://localhost:8080/servlettest/First-->404

http://localhost:8080/jsptest/FirstServlet--**>404**

http 프로토콜 정한 규칙 오류코드 - 404

web client=브라우저	web server+web application server ==> tomcat
html css js jquery	jsp servlet -자바구현 asp php cgi – 다른 언어 구현 *.py -파이썬구현

- 서블릿

1>자바 클래스

1-1. 상속 클래스 – extends HttpServlet

1-2. HttpServlet 메소드 (작업내용에 따라서) doGet() 기본

생명주기 메소드

init()

doGet()- 요청 – 처리 – 응답

destroy()

2> 웹서버 존재하며 url 요청 – 서버 실행 결과 브라우저 전송 응답

3> 서블릿 실행 결과 브라우저 해석 실행 가능 -> html css js ...

4> url 구조

http:ip:port/컨텍스트명/서블릿url매핑이름

http:ip:port/컨텍스트명/abc	http:ip:port/컨텍스트명/abc
@WebServlet("/abc") class A extends HttpServlet{} @WebServlet("/abc")- 3.0이후(현재4.0) class C extends HttpServlet{} 1> WebServlet annotation설정	package test; class B extends HttpServlet{} 2> web.xml설정 – 이전버전부터(1.0) <servlet> <servlet-name>b</servlet-name> <servlet-class>test.B</servlet-class> </servlet> <servlet-mapping> <servlet-name>b</servlet-name> <url-pattern> /abc</url-pattern>

	</servlet-mapping>

- 서블릿 실행 과정

1> 클라이언트1가 요청 - 서버 시작후 서블릿 최초 호출

2> 서블릿 객체 생성- 메모리 로드 – init 호출

3> doGet() 호출 – 응답 전송

4> 클라이언트2가 요청 - 서버 시작후 서블릿 최초 호출 아니다

서블릿 객체 생성- 메모리 로드 – init 호출

5> > doGet() 호출 – 응답 전송

6> 서블릿 메모리 삭제- (tomcat 종료시점이나 서블릿 코드 재컴파일)

1개 – 클라이언트 여러개 – 서버 메모리 객체 1개

7> 서블릿 멀티스레드 동작-내장

6장