**1/8 (월)**

**1) <HTML>**

HTML : HyperText Markup Language

HyperText : 링크가 있는 텍스트

Markup Language : 페이지에 더 많은 일을 할 수 있게 함 ex. 이미지, 링크, 표, 목록 등

2) 항상

<!DOCTYPE html>

<html>

</html>

기본 골격이다.

3) 태그 : <>내부의 것. 항상 쌍으로 표시됨 (시작 태그, 종료 태그)

4) 머리를 만들어라.

<head></head>

html파일에는 항상 머리와 몸의 두 부분이 있다.

머리에는 제목과 같은 HTML 파일에 대한 정보가 들어가 있다.

5) 몸을 만들어라

<body></body>

6) paragraphs and headings

body안에 여러 개 단락을 만들 수 있음 <p> </p>

<h1></h1>은 가장 큰 글자 크기

7) headings에 대해 더 알아보기

<h1>~~~~ <h6> 까지 있음

1. HTML is used to give websites structure.
2. We open HTML files using a browser, and the browser **renders**(shows us) the file.
3. HTML files have a <head> and a <body> (just like you!)
4. In the head, we have the <title>tags, and we use these to specify the webpage's name.
5. How to make headings and paragraphs.

8) 링크 걸기

<a href=” [https://www.codecademy.com](https://www.codecademy.com/)”>설명쓰기</a>

9) 이미지 걸기

<img src=”링크” />

10) 이미지 누르면 링크로 타고 들어가기

<a href=”링크”>

<img src=”링크” />

</a>

JavaScript is the programming language of the Web. The overwhelming majority of

modern websites use JavaScript, and all modern web browsers—on desktops, game

consoles, tablets, and smart phones—include JavaScript interpreters, making JavaScript

the most ubiquitous programming language in history. JavaScript is part of the

triad of technologies that all Web developers must learn: HTML to specify the content

of web pages, CSS to specify the presentation of web pages, and JavaScript to specify

the behavior of web pages. This book will help you master the language.

If you are already familiar with other programming languages, it may help you to know

that JavaScript is a high-level, dynamic, untyped interpreted programming language

that is well-suited to object-oriented and functional programming styles. JavaScript derives its syntax from Java, its first-class functions from Scheme, and its prototype-based inheritance from Self. But you do not need to know any of those languages, or be familiar with those terms, to use this book and learn JavaScript.

The name “JavaScript” is actually somewhat misleading. Except for a superficial syntactic

resemblance, JavaScript is completely different from the Java programming language.

And JavaScript has long since outgrown its scripting-language roots to become

a robust and efficient general-purpose language. The latest version of the language (see

the sidebar) defines new features for serious large-scale software development.

**1/9 (화)**

**What can we use JavaScript for?**

* make websites respond to user interaction
* build apps and games (*e.g.*blackjack)
* access information on the Internet (*e.g.* find out the top trending words on Twitter by topic)
* organize and present data (*e.g.*automate spreadsheet work; data visualization)

console.log(3\*4) //출력

팝업창을 띄울 수 있는 것 – confirm, prompt

confirm(“홈페이지를 종료할거임?”) //웹 사이트에서 JS가 상호작용할 수 있는 방법에 대한 예시

prompt(“What is your name?”) //입력을 요청할 수 있음

~~~.length //띄어쓰기 포함

“Hello”.substring(0,3) //=>Hel 나온다. 0번째~2번째 문자가 나오고 3번째자리에서 잘림

**1/10 (수)**

자바스크립트는 정수 값과 실수 값을 구분하지 않음.

모든 숫자를 실수로 표현한다.

16진수의 경우 0x나 0X뒤에 16진수 숫자들이 뒤따르는 형태.

복잡한 산술 연산은 Math 객체를 통해 지원된다.

3.1.5 날짜와 시간

Date() 생성자를 제공한다 //날짜와 시간을 표현하는 객체를 생성하기 위함.

var later=new Date(2010,0,1,17,10,30); //2010년 1월 1일, 오후 5시 10분 30초

later.getFullYear() //=>2010

later.getMonth() //=>0 : 월은 0부터 시작한다.

3.2 텍스트

문자열은 16비트 값들이 연속적으로 나열된 것으로, 각 문자는 수정할 수 없는 유니코드 문자로 표현된다.

클라이언트 측 자바스크립트 프로그래밍에서 자바스크립트 코드는 HTML 코드를 포함할 수 있고, HTML코드도 자바스크립트 코드를 포함할 수 있다. 자바스크립트처럼 HTML은 문자열을 구분하기 위해 작은따옴표 혹은 큰따옴표를 사용한다. 따라서 자바스크립트와 HTML을 섞어서 사용할 때는 자바스크립트 코드에서 작은따옴표나 큰따옴표 중 하나를 사용하고 HTML코드에서 나머지 하나를 사용하는 것이 좋다. ex) <button onclick=”alert(‘Thank you’)”>Click Me</button>

3.2.2 문자열 리터럴의 이스케이프 문자열

역슬래시 문자를 이용하기

‘You\’re right’

3.5 전역 객체

전역 객체의 프로퍼티는 자바스크립트 프로그램 전역에서 사용할 수 있게 정의된 심벌이다. 스크립트 인터프리터가 시작할 때 (혹은 웹브라우저가 새로운 페이지를 불러올 때), 새로운 전역 객체를 만들고 그 프로퍼티들을 초기화한다.

3.6 레퍼 객체

자바스크립트 객체는 복합적인 값이다. 객체 안의 값들은 프로퍼티 또는 이름있는 값들의 집합이며 표기법을 사용하여 프로퍼티의 값을 참조한다. 프로퍼티의 값이 함수일 때, 이 함수를 메서드라고 부른다.

var s=”hello world”;

var word=s.substring(s.indexof(“ “)+1, s.length); //문자열 프로퍼티 사용

왜 문자열은 객체도 아닌데 프로퍼티를 가지고 있을까?

프로퍼티에 어떤 값을 할당한다면 무시될 것이다. 값을 할당하는 것은 임시 객체에서 수행되며, 지속되지 않는다. (임시 객체가 회수됨) 문자열, 숫자, 불리언의 프로퍼티에 접근하려고 할 때 생성되는 임시 객체는 레퍼 객체로 알려져 있다. 문자열, 숫자, 불리언 값의 프로퍼티가 읽기 전용이고, 이 값들에 새로운 프로퍼티를 정의할 수 없다는 사실과 이 값들이 객체와 다르다는 사실을 알아야 한다.

3.7 원시 타입과 참조 타입

원시 타입 (undefined, null, 불리언, 숫자, 문자열)과 참조 타입 (배열과 함수를 포함)의 차이점 :

원시 타입의 값은 수정할 수 없다.

다시 말해 값의 본래 타입을 바꿀 수 없다.

문자열의 경우 특정 배열의 원소를 변경하지 못한다.

3.8.2 명시적 변환

Boolean(), Number(“”), String(), Object(3) //=>new Number(3)

**1/12 (금)**

var A=Math.random() //=>0에서 1 사이에 값 랜덤으로 받아진다.

var 구문에서 변수에 초기 값을 지정하지 않는다면, 변수는 값이 설정될 때까지 undefined 값을 갖게 된다. C나 자바 같은 정적 타입의 언어를 다룬 적이 있다면, 자바스크립트 변수 선언에는 타입을 명시하지 않음을 알 수 있다. 처음에 숫자를 할당하고 나중에 그 변수에 문자열을 할당해도 문법적으로 문제가 없다.

3.10.1 함수 유효범위와 끌어올림 (Hosting)

블록단위의 유효범위가 아닌 함수 유효범위를 갖는다.

function test(o) {

var i=0;

if (typeof o==”object”) {

var j=0;

for (var k=0;k<10;k++) {

console.log(k);

}

console.log(k); //k는 여전히 정의되어 있다!!!!!!

}

console.log(j); //j는 정의되어 있다.

}

var scope=”global”;

function f() {

console.log(scope); //”global”이 아니라 “undefined”를 출력한다. (충격)

var scope=”local”; //”변수를 여기서 초기화하지만 전역적으로 정의된다.

console.log(scope); //”local”을 출력한다.

}

함수 유효범위 규칙 때문에,

지역 변수는 함수 전체에 걸쳐서 정의되어, 같은 이름의 전역 변수는 함수 전체에서 이 지역 변수에 의해 감춰진다.

* 함수 맨 꼭대기에 변수를 선언할지도~

3.10.2 프로퍼티로서의 변수

전역 자바스크립트 변수를 선언할 때, 실제로는 전역 객체의 프로퍼티를 정의하는 것이다!

변수를 선언하려고 var를 사용한다면 생성된 프로퍼티는 수정 가능하지 않고 이것은 delete 연산자로 소멸시킬 수 없다는 뜻이다. 선언하지 않은 변수에 값을 지정하려고 할 때 (var 없이) 자동으로 전역 변수를 생성한다. 이 경우 전역 객체의 평범하고 수정 가능한 프로퍼티이며 삭제 가능하다.

4.4 프로퍼티 접근 표현식

① 표현식.식별자 // (표현식 => 객체), (식별자 => 객체에서 찾고자 하는 프로퍼티 이름)

② 표현식[표현식] //(표현식 => 객체, 배열), (표현식 => 객체의 특정 프로퍼티 이름이나 배열 내 원소의 인덱스 값)

4.13.2 typeof 연산자

피연산자의 데이터 타입을 가리키는 문자열이다.

ex) null => “object”

var coinFace = Math.floor(Math.random() \* 2);

while(coinFace === 0){

console.log("Heads! Flipping again...");

var coinFace = Math.floor(Math.random() \* 2);

}

console.log("Tails! Done flipping.");

1/15 (월)

http://www.example.com:80/services?category=2&kind=patents#n10

* URL스킴 : URL에 사용된 프로토콜
* 호스트명 : 웹 서버의 호스트명, 도메인명 또는 IP주소
* 포트번호 : 웹 서버 내의 서비스 포트번호. http는 80, https는 443
* 경로 : 파일이나 애플리케이션 경로
* 쿼리스트링 : 질의 문자열, 앰퍼샌드(&)로 구분된 키=값 쌍 형식
* 프라그먼트 : 문서 내의 앵커 등 조각을 지정

1.4.1 URL을 바라보는 측면

URL은 웹 클라이언트에서 호출한다는 시점에서 보면, 웹 서버에 존재하는 애플리케이션에 대한 API(Application Programming Interface)라고 할 수 있다.

1.4.2 간편 URL

특수 문자 등을 제거하고 간결하게 만드는 방식

1.4.3 파이썬의 우아한(elegant) URL

url(r’^articles/(\d{4})/(\d{2})/(\d+)/$’, views.article\_detail),

^(패턴의 시작)$(패턴의 끝)

r : 이스케이프되지 않는 raw 스트링임을 표시

\d{4} : 4자리 숫자를 의미

\d+ : 1자리 이상의 숫자를 의미

만일 URL이 /articles/2015/03/1/ 이라면, 위 URL 패턴과 매칭이 되어 아래와 같이 인자를 넘겨주어 뷰 함수를 호출함.

views.article\_detail(request,’2015’,’03’,’1’)

1.5 웹 애플리케이션 서버

1. 웹 서버 : 웹 클라이언트의 요청을 받아서 요청을 처리하고, 그 결과를 웹 클라이언트에게 응답함. 주로 정적 페이지인 HTML, 이미지, CSS, 자바스크립트 파일을 웹 클라이언트에 제공할 때 웹 서버를 사용한다. 만약 동적 페이지 처리가 필요하다면 웹 애플리케이션 서버에 처리를 넘긴다. 그 외에도 캐시 기능, 프록시 기능 등 제공한다. 또한 다수의 클라이언트로부터 동시에 요청을 받아 처리해야 해서 동시에 접속을 허가하는 클라이언트 수의 제한 및 처리 프로세스의 관리, 요청 및 응답에 관한 로그의 기록, 안정성 확보를 위한 인증 제어 및 암호화 처리 등 HTTP/HTTPS의 제어에 필요한 여러 가지 기능을 제공한다.
2. 웹 애플리케이션 서버 : 웹 서버로부터 동적 페이지 요청을 받아서 요청을 처리하고, 그 결과를 웹 서버로 반환한다. 주로 동적 페이지 생성을 위한 프로그램 실행과 데이터베이스 연동 기능을 처리한다. 대다수는 웹 클라이언트로부터 직접 요청을 받아 처리하는 웹 서버의 기능을 제공한다. (대규모 사이트는 X)

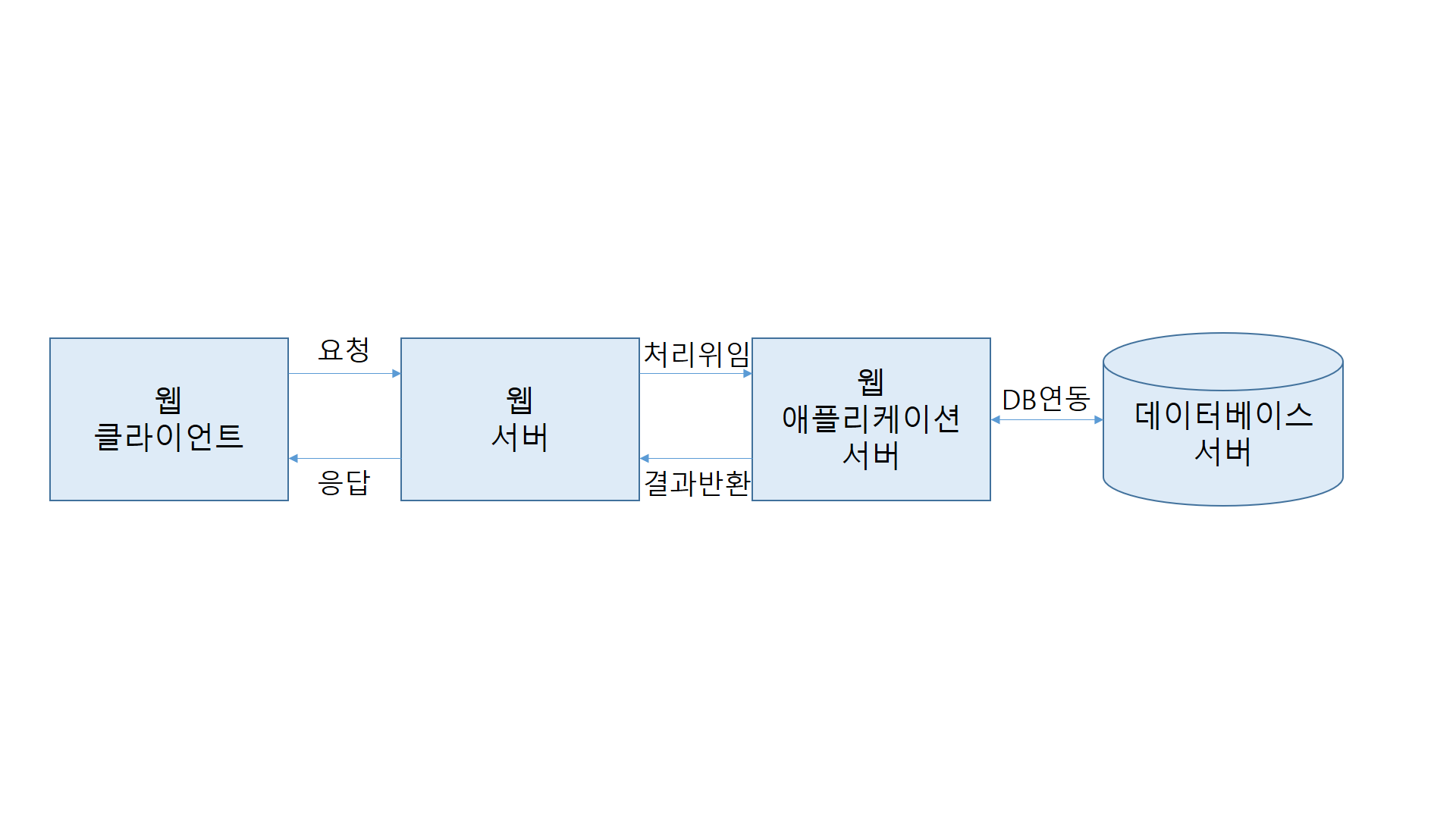
▶ 역할 분담을 하는 것이 좋음 (서버 자원 소요량 다름)

1.5.1 정적 페이지 vs 동적 페이지

- 정적 페이지란, 누가 언제 요구하더라도 항상 같은 내용을 표시하는 웹 페이지를 말함. 정적 페이지들은 해당 웹 서비스의 제공자가 사전에 준비하여 서버 측에 배치한 것으로, 동일한 리소스(URI)의 요청에 대해서는 항상 동일한 내용의 페이지를 반환한다.

- 동적 페이지란, 동일한 리소스의 요청이라도 누가 언제 어떻게 요구했는지에 따라 각각 다른 내용이 반환되는 페이지를 말한다.

1.5.4 애플리케이션 서버 방식

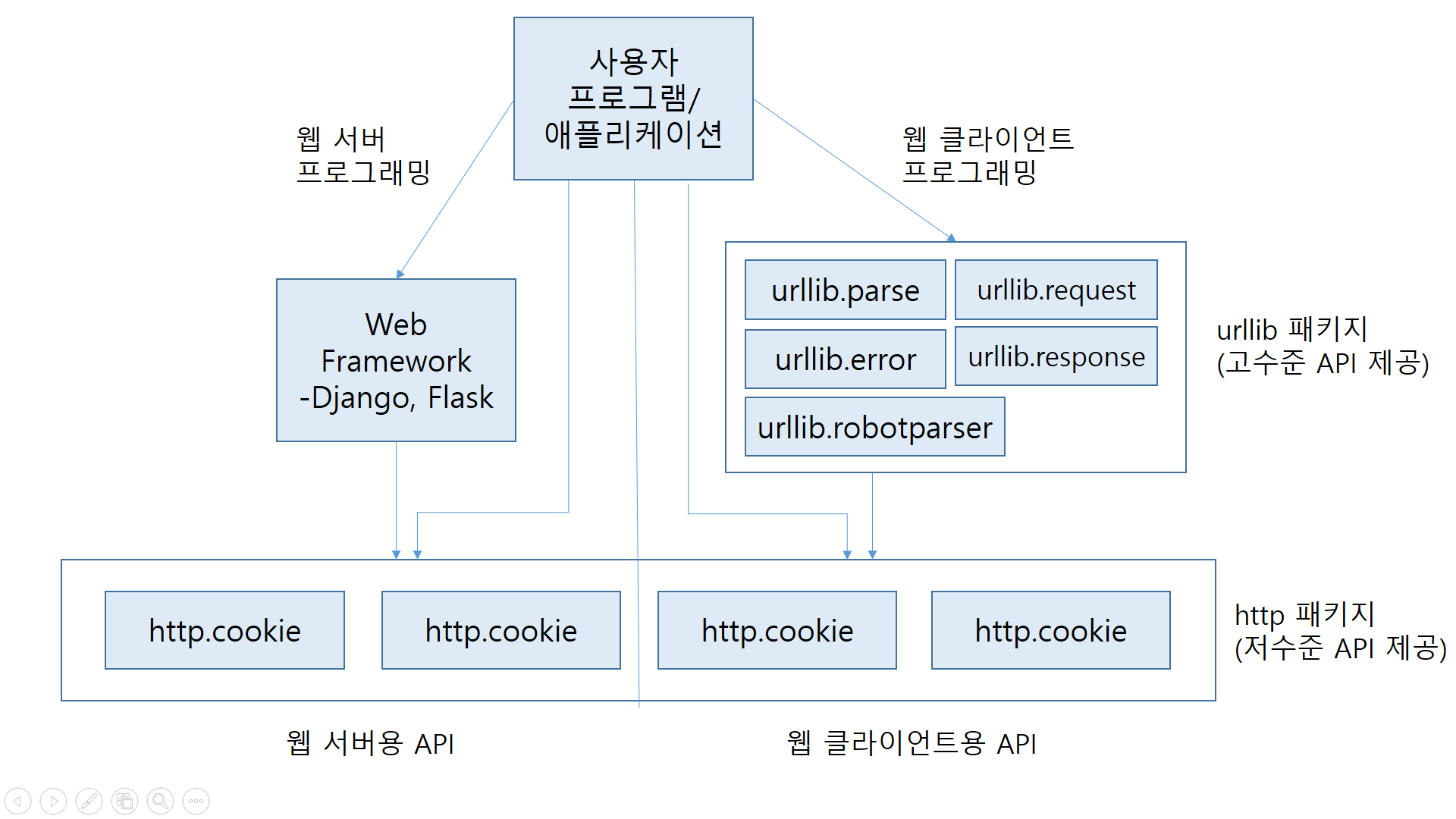


∆ SW측면의 서버 프로그램을 의미

HW측면의 용어는 웹 서버 박스 및 웹 애플리케이션 서버 박스라는 용어를 사용.

2.1 웹 라이브러리 구성

웹 프레임워크는 사용자 프로그램과 저수준의 http.serer 라이브러리 중간에 위치하여 웹 서버의 개발을 좀 더 편리하게 해주면서, 표준 라이브러리의 기능을 확장해주는 역할을 하고 있다. (파이썬의 표준 라이브러리를 사용하여 개발됨)



2.2 웹 클라이언트 라이브러리

2.2.1 urlparse 모듈

URL의 분해, 조립, 변경 등을 처리하는 함수를 제공한다. urlparse() 함수는 URL을 파싱한 결과로 ParseResult 인스턴스를 반환한다.

2.2.2 urllib2 모듈

주어진 URL에서 데이터를 가져오는 기본 기능을 제공한다. 가장 기본이 되는 urlopen()함수의 형식은 다음과 같다 :

urlopen(url, data=Nome, [timeout])

요청 방식을 POST로 보내고 싶으면 data 인자에 질의 문자열을 지정해주면 된다.

다음은 네이버 사이트에 접속하는 가장 간단한 웹 클라이언트 프로그램이다.

>>from urllib2 import urlopen

>>f=urlopen(“http://www.naver.com”)

>>print f.read(500)

위는 브라우저의 주소창에 www.naver.com 이라고 입력하는 것과 동일한 데이터를 웹 서버로부터 가져온다. 브라우저는 HTML형식의 데이터를 해석해서 화면에 보기 좋게 보여주고, 파이썬 프로그램(웹 클라이언트 프로그램)은 해석하지 않고 그대로 보여준다는 차이가 있다. HTTP GET방식을 디폴트로 사용하여 웹 서버에 요청을 보낸다.

POST방식으로 웹 서버에 요청을 보내는 방법?

urlopen() 함수를 호출 시 data 인자를 지정해주면 된다. data인자는 URL 인코딩된 문자열이다.

>>from urllib2 import urlopen

>>data=”query=python”

>>f=urlopen(“http://www.naver.com”, data)

>>print f.read(300)

3. Django 웹 프레임워크

3.1 일반적인 특징

-MVC 패턴 기반 MTV(Model-Template-View)

View -> 데이터를 가져오고 변형함

Template -> 데이터를 사용자에게 보여줌

-캐시시스템

-자체 템플릿 시스템

-테스트용 웹 서버를 포함하고 있어서 아파치 등의 상용 웹 서버가 없어도 테스트를 진행할 수 있다.

-소스 변경사항 자동 반영

**장고란 무엇인가요?**

Django(*/dʒæŋɡoʊ/ jang-goh/쟁고/장고*)는 파이썬으로 만들어진 무료 오픈소스 웹 애플리케이션 프레임워크(web application framework)입니다. 쉽고 빠르게 웹사이트를 개발할 수 있도록 돕는 구성요소로 이루어진 웹 프레임워크랍니다.

웹사이트를 구축할 때, 비슷한 유형의 요소들이 항상 필요합니다. 회원가입, 로그인, 로그아웃과 같이 사용자 인증을 다루는 방법이나 웹사이트의 관리자 패널, 폼, 파일 업로드와 같은 것들 말이지요.

그런데 정말 다행이게도, 오래전에 어떤 웹 개발자들이 새로운 웹 사이트를 개발할 때 서로 비슷한 문제들에 직면한다는 것을 깨달았습니다. 그래서 팀을 조직했고요. 바로 사용할 수 있는 구성요소들을 갖춘 여러 프레임워크를 만들었답니다. 장고도 그중에 하나인 거죠. 다시 발명해야 하는 문제로부터 해방감을 주고요. 새로운 웹사이트를 개발할 때 뒤따르는 간접비용의 부담을 덜어준답니다.

## 왜 프레임워크가 필요한가요?

장고라는 것이 실제로 무엇을 위한 것인지 이해하기 위해서는 서버에 대해서 자세히 살펴볼 필요가 있어요. 먼저 서버가 여러분에게 웹 페이지를 제공하길 원하는지 알아야 해요.

편지(request, 요청)가 도착했는지 확인해주는 메일박스(port, 포트)가 있다고 상상해보세요. 이 것은 웹 서버가 해주는 일이에요. 웹 서버는 받은 편지를 읽고 웹 페이지와 함께 답장을 준답니다. 그런데 무언가를 주고 싶을 때는 그 안에 내용이 있어야 하죠. 장고는 그 특정 콘텐츠를 만들 수 있는 역할을 합니다.

## 누군가가 서버에 웹 사이트를 요청하면 어떤 일이 벌어질까요?

웹 서버에 요청이 오면 장고로 전달됩니다. 장고 **urlresolver**는 웹 페이지의 주소를 가져와 무엇을 할지 확인합니다.(urlresolver는 웹 사이트 주소인 URL(Uniform Resource Locator)를 통해 이해합니다). 이 urlreslover는 그리 똑똑하지 않습니다. 패턴 목록을 가져와 URL과 맞는지 처음부터 하나씩 대조해 식별합니다. 만약 일치하는 패턴이있으면, 장고는 해당 요청을 관련된 함수(view)에 넘겨줍니다

집배원을 생각해보세요. 집배원은 거리를 걸으며 집집이 편지와 대조해서 주소와 번지를 확인합니다. 주소와 번지가 일치하면 그곳에 편지를 배달합니다. urlresolver가 바로 집배원과 같은 역할을 합니다.

모든 재미난 일들은 view 함수에서 처리됩니다: 특정 정보를 데이터베이스에서 찾을 수 있습니다. 그런데 만약 사용자가 데이터를 바꿔달라고 수정을 요청한다면 어떻게 될까요? "제 직업에 대한 설명을 바꿔주세요."와 같은 편지를 받았다고 생각해봅시다. view함수는 수정할 수 있는 권한이 있는지 확인하고 나서, 직업에 대한 설명을 수정해 다시 답장을 주겠지요. "완료했습니다!" 라고요. 그러고 나서 view는 답장을 생성하여, 장고는 그 답장을 그 사용자의 웹 브라우저에 보내주는 역할을 합니다.

물론 지금까지의 설명은 아주 간단히 설명한 것에 불과해요. 하지만 모든 기술적인 부분까지 자세히 알 필요가 없답니다. 이 정도 아는 것만으로도 충분하답니다.

좀 더 자세히 아는 것보다, 지금부터는 장고를 이용해 간단한 것부터 만들어봐요. 그렇게 하면서 중요한 모든 것들을 하나씩 배워나가도록 해요.