

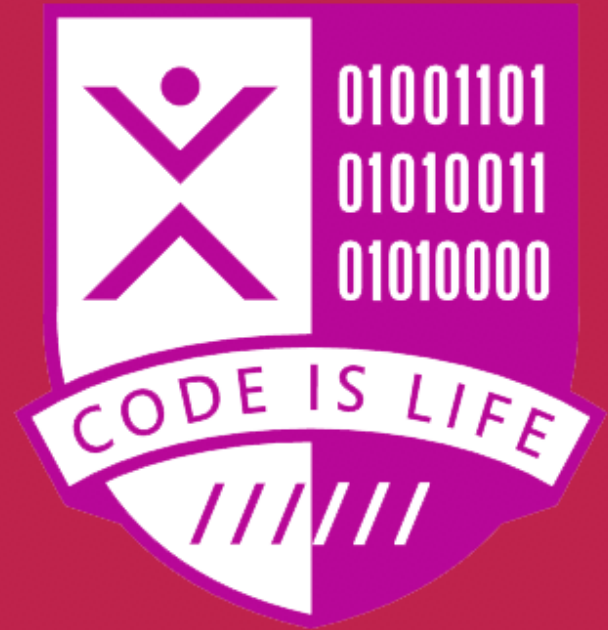
MSP evangelism 2016

장고: 분노의 개발자

사전과제 3

html

Team. 방울뱀





What is markup language?

머리말입니다↵

[머리말(양 쪽)] hello world!↵

mspdjang2016↵

[사각형] 안녕 하세요↵



Microsoft

[그림]↵

한글의 조판부호 보이기

```
<html>
  <head>
    <title>page title</title>
  </head>
  <body>
    <h1> this is a heading</h1>
    <p> this is a paragraph</p>
  </body>
</html>
```

html



Microsoft



Then, what is DOM?



잠시 다른 이야기 !

Tree structure란?

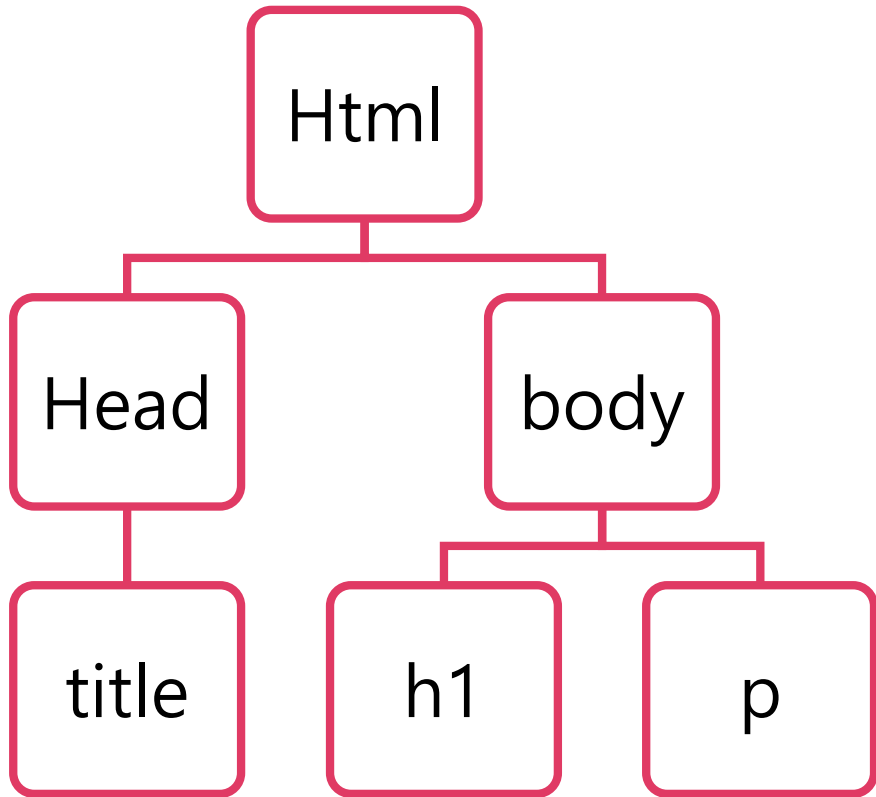
- 정보를 어떻게 저장할까 고민해 만든 구조
- Parent-Child 구조이므로 Directed & Acyclic
 - 모든 Node에 대해 부모와 자식이 있고,(비어 있을 수 도 있음)
 - 자식은 직속 부모가 단 하나
 - 부모는 여러 자식을 가질 수 있다.
 - 자식도 작은 Subtree이다. (재귀적인 구조)
 - Tree가 강력한 이유 : 일관적이다.



잠시 다른 이야기 !

앞에서 봤던 html의 구조를 tree로 바꿔보았다.

왜, tree 구조를 사용하는가?



- One object에 대해 Multiple attributes를 넣기 편해진다.
 - 매 속성마다 새로운 tag를 만드는 대신 Tree 구조를 이용하면 간편해짐.
 - Ex) 진하게 + 이탤릭체를 하려면 `<bi>` tag를 새로 만드는 대신 `<i></i>`로 해결한다.
- Insertion & Deletion이 용이하다.
- Tree 구조가 컴퓨터 공학에서 많이 사용되기 때문에, 구현이 용이하다.



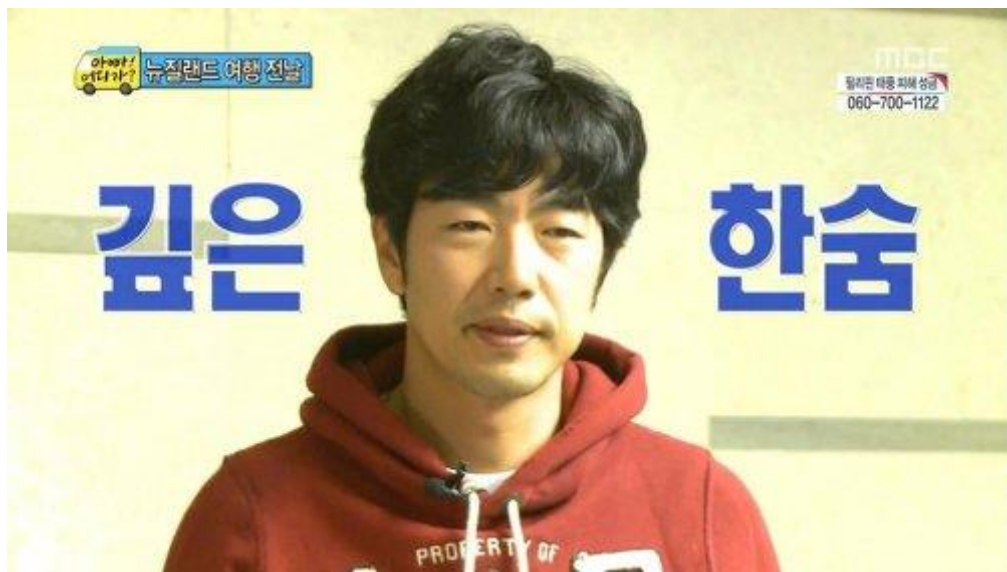
다시 DOM으로 돌아와서, DOM이란?



DOM

Document Object Model

- Programming **API** for documents



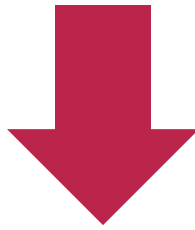
API 란..?



DOM

Application Programming Interface

- 응용프로그램에서 사용할 수 있도록,
- 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스



어떻게 돌아가는지 몰라도 시키는 대로만 짜면 돌아갑니다.
Ex) 진한 글씨를 원하면 `` tag를 쓰세요 !



DOM

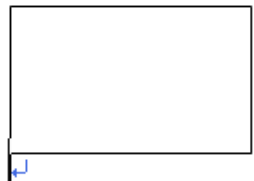
Document를 예쁘게 만들려면, 정해진 규칙대로 쓰면 됩니다.

머리말입니다 ↴

[머리말(양 쪽)] hello world! ↴

mspdjangoo2016 ↴

[사각형] 안녕하세요 ↴



[그림] ↴

다시 한글의 조판부호를 봅시다.

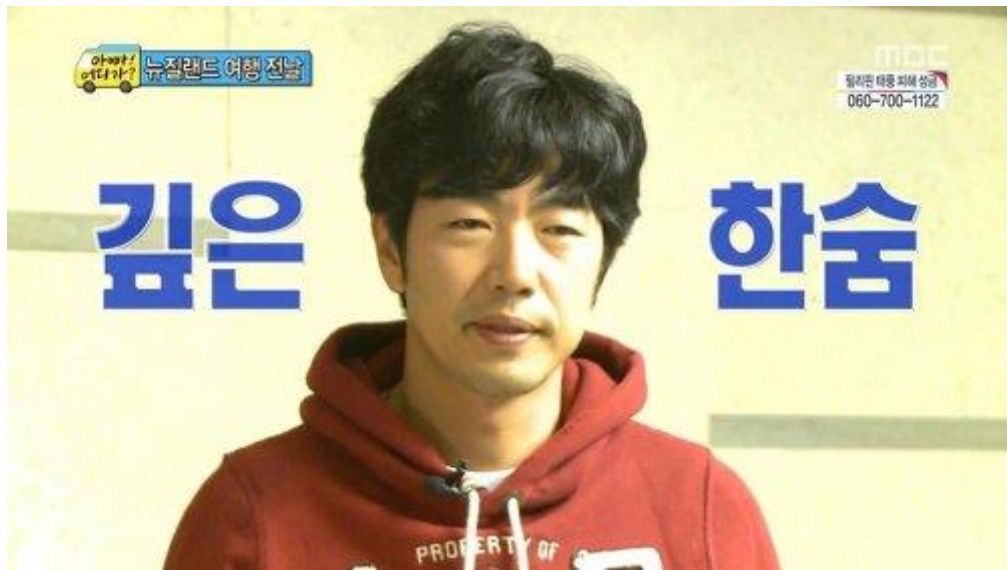
우리가 직접 볼 일이 없을 뿐,
실제로 처리를 다 해줍니다.

API의 API인 셈이죠!
(DOM을 사용자UI로 대체한 셈)



Document Object Model

- Programming API for documents
- Document의 각 내용을 Object로 치환하는 것



Object 란..?



DOM

Object

- 현실에서 지각 가능한 물체들을
- 컴퓨터 내부에서 처리 가능하도록
- 유한의 값으로 치환해준 것
 - Ex) Car object
 - 명사 : color, size, model, ... -> Variables(변수)
 - 동사 : run, stop, turn, ... -> Procedures(방법)



DOM

DOM으로 다시 돌아와서,

- Document Object의 경우
- Procedure는 없고 오직 Variable만 있음.
- 두가지 종류의 Variable이 있다.
 - 1. 해당 Object가 무엇을 보여줄지를 결정하는 Variable
 - 이들을 우리는 Value라고 부른다.
 - Ex) tag사이에 들어가는 `이것!`
 - 2. 해당 Object가 어떻게 보일지를 결정하는 Variable
 - 이들을 우리는 Attributes라고 부른다.
 - Ex) `<a>` tag의 href



구체적인 예시 살펴보기

<html>, <head>, <body>

- 무조건 존재하는 HTML의 기본 구조
- 하지만 예시처럼 사실 없어도 돌아가긴 한다.
- 알아서 html body를 넣어주는 셈이기 때문

<h1>, <h2>, ... <h6>, <p>

- 문단 (Paragraph) 나누기, 제목 (Header) 만들기.



구체적인 예시 살펴보기

 or ,

- : Unordered List
- : Ordered List
- : List의 각 element들

<form>, <input>

- 어떠한 폼으로 사용자 입력 받기
- Ex) 설문조사 "form", 로그인 "form"



구체적인 예시 살펴보기

<div>,

- : Value에서 고치고 싶은 게 생기면(inline elements)
- <div> : 여러 Object들을 정리하기 위함.

그리고, 대망의 <a>

- HTML 이름이 Hyper Text Markup Language 인 이유
- href(attribute) : hypertext reference,
다른 페이지로 넘어갈 수 있도록!



추가로, Style attribute

- 모든 tag가 갖고 있는 attribute
- ex) style = "color : red" -> 해당 tag의 색이 빨강이 됨.
- 원래 이럴 줄은 몰랐는데, 점점 발전하면서
이 attribute에서 처리해야 하는 양이 너무 많아 짐!
 - Cascading Style Sheet라고 하는 언어 아닌 언어가 생겨나버렸다.
 - Style attribute에 들어가야 하는 내용만 빼서 정리
 - Ex) h1{color : red ; text-align : center;}
 - 요즘에는 HTML에서 직접 style을 고치는 경우는 드물다.
 - Tag, Class Id selector
 - Tag : 지금까지 본 것들
 - Class : class = "" Attribute, 여러 tag가 가질 수 있다.
 - Ex) p.center{} or . center{}
 - Id: id = "" Attribute, 오직 한 tag만 가질 수 있다.
 - Ex) #id1{}



And, much more !

www.w3schools.com/tags/



<a> tag가 나온 김에, 인터넷 구조를 잠시 알아보자.

- href = "http://www.naver.com을 너무 당연하게 받아들이는 건 아닐까 ?
- 인터넷은, 그러니까 우리가 사용하고 있는 것들은
- 다 어떤 방식으로든 연결 되어있다.
- 우리는 NAVER가 갖고있는 컴퓨터에 접근하는 셈이다.
 - User의 Request, Server의 Response
- 결국 Web은 수많은 컴퓨터가 연결 되어있는 상태 그 자체를 말한다.



그리고, 한 가지 더 !

- HTML은 정적(Static)이다
- 언급한 대로 Procedure가 없기 때문에.
- 하지만 사람들은 더 예쁜 사이트를 요구했다.
- <link>, <script>를 통해 Procedure가 있는 language를 연결해서 쓰자 !!
 - Cf. 사실 link로 CSS도 연결해서 쓴다.
- javascript가 대표적으로 쓰인다.
- jQuery를 통해 DOM Tree를 동적으로 바꾸고
- Ajax를 통해 Request를 하고 Response를 해보자 !