1. 속성(Attribute)과 요소(property)의 차이가 무엇인가요?

**attribute**

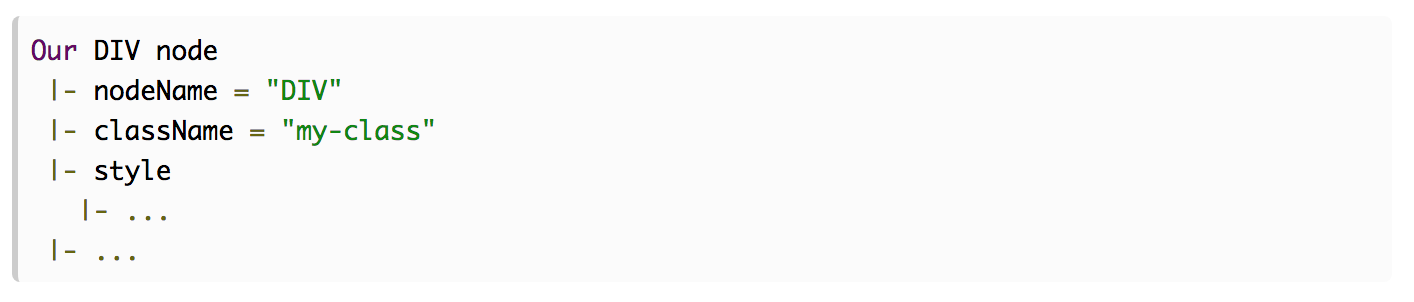
attribute 는 html 문서에서 elements 에 추가적인 정보를 넣을 때 사용되는 요소입니다.

<div class="my-class"></div>

위의 예시에서 class는 속성(attribute)이고, my-class는 속성값(value)이 된다.

**property**

property는 html DOM 안에서 attribute 를 가리키는(혹은 대신하는) 표현입니다.



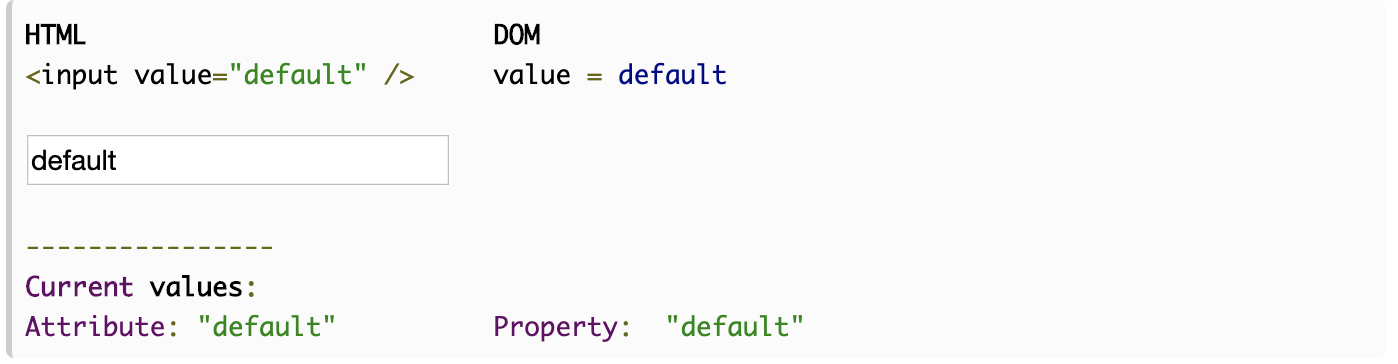
위 내용은 <div class= ‘my-class’></div> 태그를 DOM 으로 표현한 것인데 위에서 className이 property가 되는 것이죠. 다시 말하면 html 문서 안에서는 class가 attribute를 의미하지만 html DOM 안에서는 property를 의미한다고 보시면 될 것 같습니다.

attribute 는 html 문서 안에서의 정적인(바뀌지 않는) 속성 그 자체를 의미하고, property 는 html DOM 안에서 동적인(바뀌는) 속성(또는 그 값)을 의미합니다.

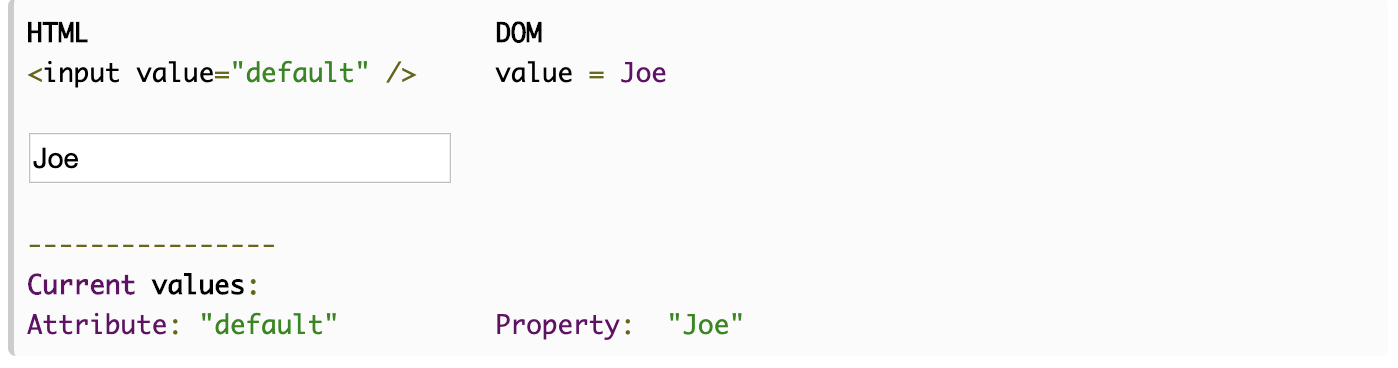
**Attribute와 Property의 차이점**

attribute는 HTML 문서 안에 있는 것이고 property는 HTML DOM tree 안에 있는 것이다. 이 의미는 attribute는 변하지 않지만 property는 변할 수 있다는 뜻이다.

예를 들어, 사용자가 체크박스를 체크하거나 input 박스에 텍스트를 입력했을 때 JavaScript로 값을 변경하면 property의 값은 변하게 된다.



위 사진에서 attribute와 property값이 default로 동일한 것을 확인할 수 있다.



하지만 input 박스에 "Joe"를 입력하게 되면 attribute값은 그대로지만 property값이 변경된 것을 확인할 수 있다.

왜냐하면 property는 DOM 안에 존재하며 동적인 반면, attribute는 HTML 문서 안에 존재하며 정적인 상태로 변하지 않기 때문이다.

2. non blocking과 blocking 을 설명하시오

**blocking**

호출된 함수가 자신의 작업을 모두 끝낼때까지 제어권을 가지고 있어 호출한 함수가

대기하도록 만든다

**non blocking**

호출된 함수가 바로 return 해서 호출한 함수에게 제어권을 주어 다른 일을 할 수 있게 함

호출되는 함수가 바로 리턴하느냐 마느냐가 중점

위 내용을 보기에 앞서 동기방식과 비동기 방식의대한 설명이 필요

**Synchronous / Asynchronous**

**Synchronous (동기)**

동기는 함수를 호출하고 호출된 함수의 작업이 완료된 후의 return 을 기다리거나

return 을 받더라도 호출한 함수가 계속해서 작업완료 여부를 신경씀

**Asynchronous (비동기)**

동기는 함수를 호출할 때 callback 함수를 같이 전달해 작업이 완료되면 callback을 실행

작업완료를 callback이 신경씀

여기서 조합을 살펴보면

Blocking &amp; Sync 와 NonBlocking &amp; Async 는 쉽게 이해가 된다.

하지만 NonBlocking &amp; Sync 와 Blocking &amp; Async 는 조금 설명이 필요하다

NonBlocking &amp; Sync 를 알아보자

NonBlocking 은 바로 return을 해서 제어권을 준다고 했고, Sync 는 작업완료여부를 호출한

쪽에서 신경을 쓴다고했다.

그림을 보면 호출을 하고 바로 반환이 되고 다른일을 수행한다.

이후에 작업이 완료되었는지 계속 물어보는 일을 추가로 수행하는것이 NonBlocking &amp; Sync 이다.

Blocking &amp; Async 를 알아보자

우선 Blocking 은 작업이 완료될때까지 제어권을 호출된 쪽에서 가지고 있고, Async 는

작업완료여부를 호출된 쪽에서 신경을 쓴다. 곱씹어보면 약간 이상할 수 있다.

어차피 제어권이 없는 상태에 결과만 기다리는 Blocking &amp; Sync 와 별 차이가 없는것 같다.

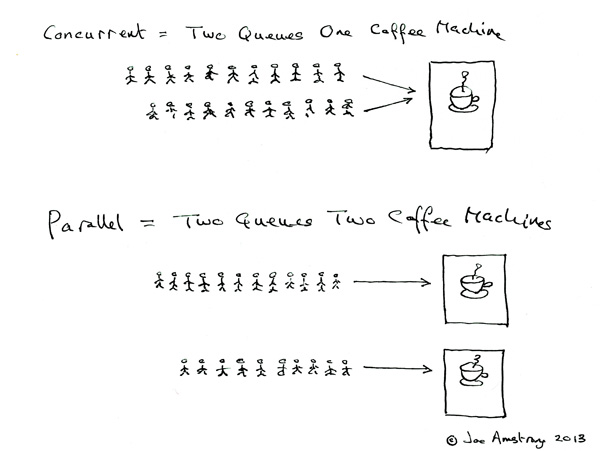
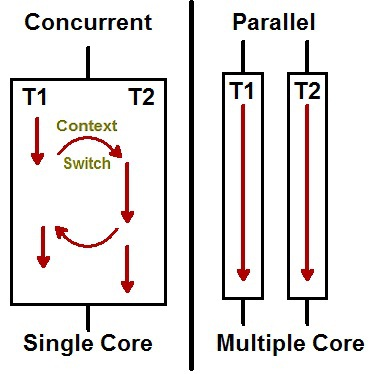
이 방식은 특별한 장점이 없어 일부러 사용할 필요는 없다고 한다.

보통 NonBlocking &amp; Async 방식을 쓰는데 그 과정중 하나라도 Blocking 이 포함이되면

의도치않게 Blocking &amp; Async 로 작동한다고한다.

3. Concurrency(동시성)과 Parallelism(병렬성)을 설명하시오.

|  |  |
| --- | --- |
| 동시성 | 병렬성 |
| 동시에 실행되는 것 같이 보이는 것 | 실제로 동시에 여러 작업이 처리되는 것 |
| 싱글 코어에서 멀티 쓰레드(Multi thread)를 동작시키는 방식 | 멀티 코어에서 멀티 쓰레드(Multi thread)를 동작시키는 방식 |
| 한 번에 많은 것을 처리 | 한 번에 많은 일을 처리 |
| 논리적 개념 | 물리적 개념 |



싱글 코어에서는 2개의 작업을 동시에 실행되는 것 처럼 보이기 위해 번갈아 가면서 작업을 수행합니다. 이때 다른 작업으로 바꾸어 실행할 때 내부적으로 Context switch가 일어납니다.