

#. 중요 HTML 태그 목록

〈a〉 → 〈a href=“ ”〉

〈div〉

〈span〉

〈ol〉

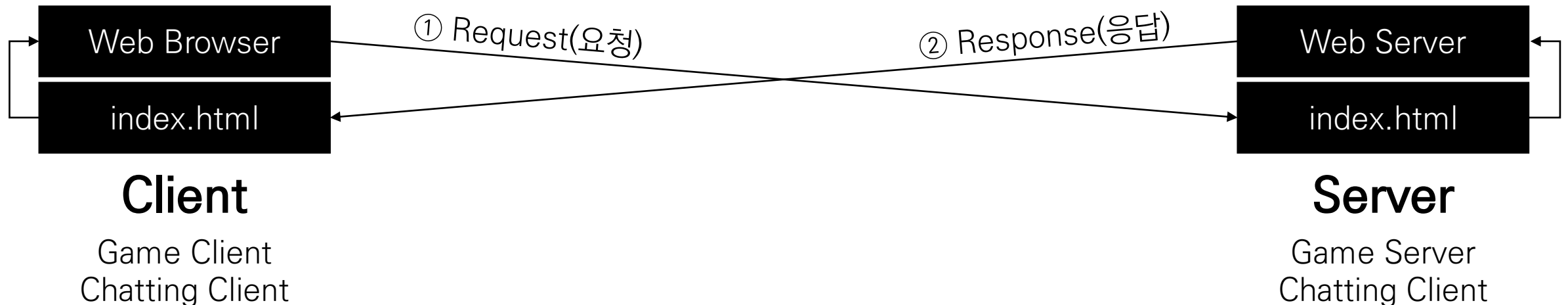
〈ul〉

〈li〉

17. 인터넷을 여는 열쇠 : 서버와 클라이언트

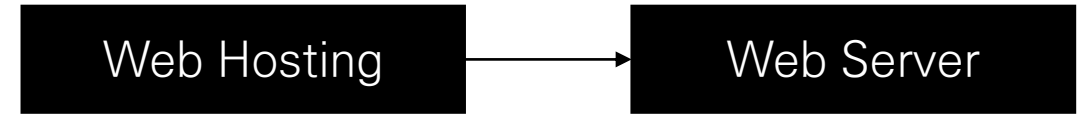
- 인터넷이 동작하는 기본적인 원리를 알아보기 위해 컴퓨터는 최소 2대가 있어야 한다.
- 각각의 컴퓨터를 역할에 따라 명칭을 정했는데, 자본주의에서 통용되는 용어로 서비스 제공자와 서비스 이용자 (Server & Client)라고 부르기로 했다. 예를 들어 게임이라면, 게임 이용자가 보는 페이지를 Game Client, 게임 회사가 보는 컴퓨터는 Game Server라고 할 수 있다.

1. Browser 컴퓨터에서 Server 컴퓨터 하드디스크에 들어있는 index.html 파일을 Request(요청)한다.
2. index.html은 Server 컴퓨터에서 Browser 컴퓨터로 Response(응답)한다.
3. 그 결과 이용자(Client)는 Browser를 통해 index.html 파일을 본다.



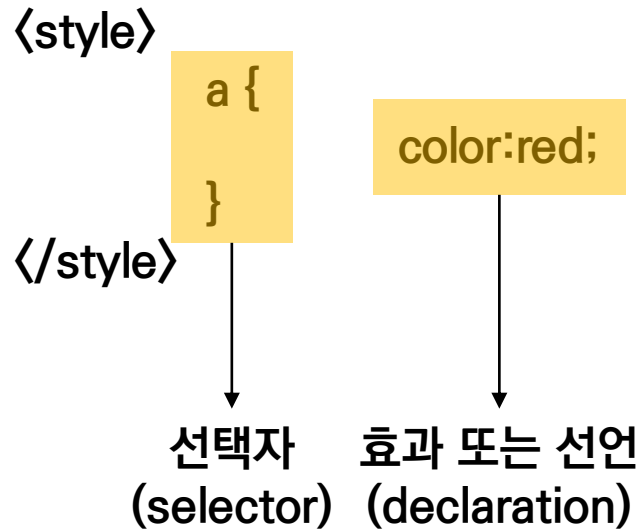
17. 인터넷을 여는 열쇠 : 서버와 클라이언트

- 인터넷이 연결된 컴퓨터에 웹 브라우저를 깔면 누구나 볼 수 있도록 환경을 만들 수 있다. 이를 위해 2가지 방법을 알아보고자 한다. (Web Hositng, Web Server)
- (첫 번째, 직접 설치, Server) 어렵다. 대신, 인터넷과 관련된 여러 가지 개념을 배울 수 있다.
- (두 번째, 업체 대행, Hosting) 쉽다. 대신, 배울 것은 적다.
- 여건에 따라 다르겠지만, Hosting → Server 순으로 학습하는 것이 효과적일 것이다.



4. CSS의 기본 문법

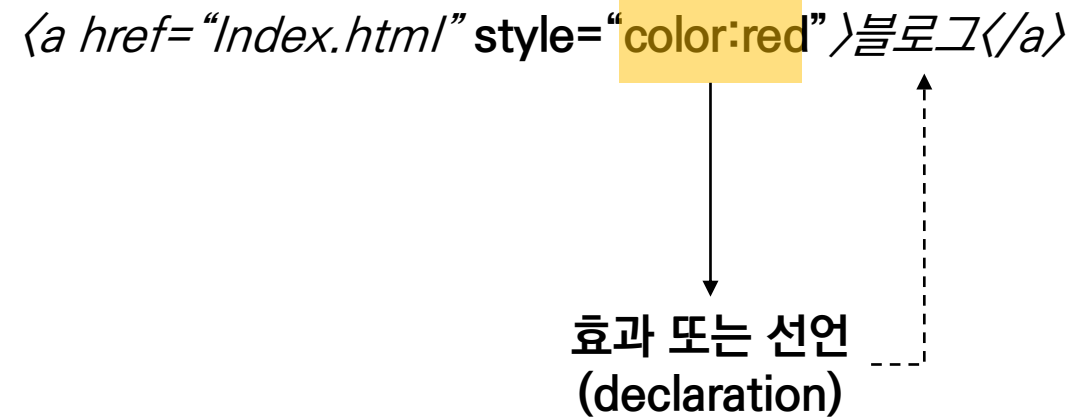
〈style〉 태그를 사용하는 경우



VS

누구를 지정하나

HTML의 style 속성을 사용하는 경우



장점

같은 성격의 태그 전체를 디자인하기 쉬움

※ 〈a〉 태그 전체를 선택자로 지정하여 디자인할 수 있음

단점

특정 성격의 태그 부분을 디자인하기 어려움

특정 성격의 태그 부분을 디자인하기 쉬움

※ '블로그'라는 텍스트의 태그에 효과를 적용하므로 선택자가 필요 없음

공통점

선언과 선언 간의 구분을 위해 세미콜론(;)을 항상 사용해야 함

3~5. HTML과 JavaScript의 만남 3가지

1. script 태그 사용

```
<script>  
document.write("Hello World")  
</script>
```

→ HTML → JavaScript

3. 콘솔(Console) 사용

Chorome 웹페이지에서 개발자도구를 사용하여 Console 도구를 연다. 해당 Console 화면에서 JavaScript 코드를 입력해서 **즉석에서 실행**한다.

(※ 문제를 가볍게 해결하고자하는 자세도 필요함)

https://www.w3schools.com/tags/ref_eventattributes.asp

2. 이벤트(Event) 사용

```
<input type='button' value='hi' onclick='alert("Hello")'>
```

↓ ↓ ↓

HTML 이벤트 JavaScript

↓

#Javascript event attributes (속성)

JavaScript

12. CSS과 JavaScript의 만남 (제어할 태그 선택하기)

`<input id="night_day" type="button" value="night" onclick="`
`document.querySelector('body');` → CSS `<body>` 태그
`>`

`<input id="night_day" type="button" value="night" onclick="`
`document.querySelector('.choice');` → CSS class
`>`

`<input id="night_day" type="button" value="night" onclick="`
`document.querySelector('#target');` → CSS Index(ID)
`>`

→ JavaScript
#JavaScript select tag by CSS selector

18. 리팩토링 중복의 제거

```
<input id="night_day" type="button" value="night" onclick="
    if(document.querySelector('#night_day').value==='night'){
        document.querySelector('body').style.backgroundColor='black';
        document.querySelector('body').style.color='white';
        document.querySelector('#night_day').value==='day';
    } else {
        document.querySelector('body').style.backgroundColor='white';
        document.querySelector('body').style.color='black';
        document.querySelector('#night_day').value==='night';
    }
">
```

앞서 만든 night/day 버튼을 화면 하단에 만들고 싶어 **복사하였으나, 제대로 작동하지 않음.**
(※ id값 'night_day'를 'night_day2'로 바꾸면 되지만, 이 역시 반복 작업이므로 좋은 방향이 아님)

18. 리팩토링 중복의 제거 (1)

```
<input id="night_day" type="button" value="night" onclick="
    if(document.querySelector('#night_day').value==='night'){
        document.querySelector('body').style.backgroundColor='black';
        document.querySelector('body').style.color='white';
        document.querySelector('#night_day').value==='day';
    } else {
        document.querySelector('body').style.backgroundColor='white';
        document.querySelector('body').style.color='black';
        document.querySelector('#night_day').value==='night';
    }
}">
```

같은 것들 동시 드래그
→ Ctrl + D

document.querySelector('#night_day')는 <input> 태그를 가리킨다.
해당 코드는 이미 <input> 안에 들어 있으므로,
자기자신을 가리키는 코드로 전환하면 훨씬 코드가 간단해진다.

18. 리팩토링 중복의 제거 (1)

```
<input type="button" value="night" onclick="
    if(this.value==='night'){
        document.querySelector('body').style.backgroundColor='black';
        document.querySelector('body').style.color='white';
        this.value==='day';
    } else {
        document.querySelector('body').style.backgroundColor='white';
        document.querySelector('body').style.color='black';
        this.value==='night';
    }
}">
```

This를 통해 코드가 훨씬 간결해졌다. 중복이 제거되었다.

18. 리팩토링 중복의 제거 (2)

```
<input type="button" value="night" onclick="
    if(this.value==='night'){
        document.querySelector('body').style.backgroundColor='black';
        document.querySelector('body').style.color='white';
        this.value==='day';
    } else {
        document.querySelector('body').style.backgroundColor='white';
        document.querySelector('body').style.color='black';
        this.value==='night';
    }
">
```

Document.querySelector('body')라는 코드도 중복

18. 리팩토링 중복의 제거 (2)

```
<input type="button" value="night" onclick="
    var target=document.querySelector("body");
    if(this.value==='night'){
        target.style.backgroundColor='black';
        target.style.color='white';
        this.value==='day';
    } else {
        target.style.backgroundColor='white';
        target.style.color='black';
        this.value==='night';
    }
">
```

이제 var target = document.querySelector('body')에서 body만 바꾸면 나머지는 자동으로 바뀌게 코드가 간결해졌고, 중복이 제거되었다.