

- 2015년전 혼합된 개념, 이후 ES6, ES6+, ES7, ES8 등을 모던 자바스크립트이라 명명

1) 링크 가상클래스와 반응 가상클래스 선택자 (CSS02_05)

CSS에서는 가상(Pseudo)이라는 개념을 적용해 HTML 문서에 실제로 존재하지 않아 접근할 수 없는 문서 모델(DOM)에 접근하여 CSS 문서를 용이하게 작성할 수 있도록 해주는 가상요소 선택자와 가상 클래스 선택자가 있다.

- 반응 가상클래스 선택자는 사용자가 마우스 포인터를 특정 요소 위에 위치 시키거나 클릭한 상황이 발생하면 그 요소를 선택해 주는 선택자 이다.
- 반응 가상클래스 선택자를 통해 스타일을 지정한 요소에 마우스 포인터가 위치된 상태이거나(hover) 클릭된(active) 상태이면 지정한 스타일이 적용되고 마우스 포인터의 위치를 요소 밖으로 이동하거나 클릭을 해제 하면 스타일은 적용되지 않는다.

태그(#id, .class)뒤에 : 콜론은 행동(Action)을 나타낸다. 주로 알고 있는 행동들은 다음과 같다.

- 태그 : 행동 { 스타일 }

```
a:link { color: blue; }
a:visited { color: darkgray; }
a:hover { color: red; } /* :link, visited 뒤에 위치해야 한다. */
a:active { color: green; }
```

- 링크(a href)의 제어 - 밑줄 제거

```
text-decoration: none;
```

- 체크박스는 체크의 유무가 (선택가능한) 행동

```
#interest_select:checked ~ #check {
  width: 500px; height: 10px; display: block;
}
```

- radio 는 그룹으로 묶이기에 여러개중 하나만 선택되는 것

관심분야 :

```
<input type="radio" name="interest_select" id="interest_select" value="interest_select"
checked="checked"/>
<label for="interest_select">선택하기</label>
<input type="radio" name="interest_select" id="interest_noselect" value="interest_noselect"/>
<label for="interest_noselect">선택하지 않음</label>
```

- select 와 noselect 에 따른 하위버튼의 display on/off

```
#interest_select:checked ~ #check {
  width: 500px;
  height: 10px;
  display: block;
}
#interest_noselect:checked ~ #check {
  display: none; }
```

기본속성이 **disable**인 **input** 박스가 있다.

체크박스를 클릭시 내용입력을 할 수 있게 활성화해주고, 일부 스타일을 적용하는 것

이메일 : ☐ 이메일 입력

이메일 : ☒ 이메일 입력

```
이메일 : <input type="text" name="email" id="email" size="30" disabled/>
      <input type="checkbox" name="input_email" id="input_email"
      onclick="checkedEmail()"/>이메일 입력<br/>
```

```
<script>

function checkedEmail() {

    // html 문서에서 id 속성의 값이 input_email인 문서 객체를 가져온다.
    var chk = document.getElementById("input_email");
    // var chk = document.querySelector("#input_email");

    /* chk는 html 문서에서 체크박스이므로 이 체크박스가 체크된 상태면 checked
    * 속성은 true가 되어 if 문의 블록({}) 안의 코드가 실행되고 체크되지 않은
    * 상태이면 false가 반환되어 else의 블록({}) 안의 코드가 실행된다.
    */
    if(chk.checked) {

        /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
        * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
        * 해제하고 배경색을 흰색으로 글자색을 검정색으로 설정한다.
        */
        document.form1.email.disabled = false;
        document.form1.email.style.backgroundColor = '#FFFFFF';
        document.form1.email.style.color = 'black';

    } else {

        /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
        * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
        * 적용하고 배경색을 회색으로 글자색을 흰색으로 설정한다.
        */
        document.form1.email.disabled = true;
        document.form1.email.style.backgroundColor = 'gray';
        document.form1.email.style.color = 'white'
    }
}

</script>
```

목록(리스트)에 특정 속성을 반복적으로 부여하기 위해 파라미터를 지정가능 하다

```
li:nth-child(2n) { background-color: #FFEAEA; }  
li:nth-child(2n+1) { background-color: #FFC9C9; }
```

background 속성의 반복적인 구문은 함축이 가능하다

- https://tcpschool.com/css/css_basic_backgrounds

```
.main_box {  
  background: url('images/meat_modum04.jpg');  
  background-repeat: no-repeat;  
  background-position: 350px center;  
  background-size: cover;  
  color: #ffffff;  
  line-height: 1.5;  
  margin: 40px auto;  
  padding: 40px;  
  width: 800px;  
}
```

```
.main_box {  
  background:  
    linear-gradient(90deg, rgba(0, 0, 0, 1) 40%, rgba(0, 0, 0, 0)),  
    url('images/meat_modum04.jpg') no-repeat 350px center;  
  background-size: cover;  
  color: #ffffff;  
  line-height: 1.5;  
  margin: 40px auto;  
  padding: 40px;  
  width: 800px;  
}
```

그라데이션은 <https://cssgradient.io/> 에서 쉽게 가져올 수 있다. //컬러피커



박스의 크기와 오버플로우

- **auto** : 넘치면 스크롤 되게(넘칠때만 스크롤바 보임)
- **scroll** : 항상 스크롤 되게(항상 스크롤바가 보임)
- **hidden** : 넘기는 컨텐츠 숨기기

박스를 다양한 유형으로 표시하기(display)

- 리스트의 . 을 없애는 것

```
list-style-type: none;
```

- **margin** 을 통해 좌우를 중앙 정렬을 **auto**로 표현

```
margin: 30px auto;
```

- 박스 영역 지정 : **display** 속성은 해당 요소를 어떻게 보여 질 것인지를 지정하는 속성

호

```
display: inline-block;
```

- **border-radius** 에도 더 강한 **value** 지정 가능

```
border-radius: 10px 50px ;
```

요소를 원하는 위치에 배치 - **position** 속성

- **Z-index** : 겹치는 부분의 우선순위

```
z-index: 1;
```

JS : 2015년6월에 대격변 ECMAScript 2015(ES2015)로 모던 자바스크립트라고 불린다.

1) Javascript01_02.html

```
<script type="text/javascript">
```

ES5 이후 **type**를 명시하지 않아도 된다. 특히 **<script>**는 어디서나 위치할 수 있게 되었다.

XML 이후 JSON 이 대세, 타입이 단조로운 단점이 있다.

<https://www.json.org/json-ko.html>

백틱으로 작성시 편하게 입력 가능

[자바스크립트 템플릿 리터럴 - 백틱](#)

```
let discription = `  
 哈哈  
`;  
document.write(discription);
```