

## □ 개념 확인

### (1) 빈 괄호를 채워 넣으시오

- ① jQuery 로 속성을 조작할 때는 ( attr() ) 메소드를 사용한다.
- ② jQuery 로 스타일 속성을 조작할 때는 ( css() ) 메소드를 사용한다.
- ③ jQuery 로 내부의 글자를 조작할 때는 ( html() ) 메소드와 ( text() ) 메소드를 사용한다.
- ④ jQuery 로 클래스 속성을 추가할 때는 ( addClass() ) 메소드를 사용한다.
- ⑤ 폼 요소의 value 값을 가져오거나 새 값을 적용할 때는 ( val() ) 메소드를 사용한다
- ⑥ 검색한 문서 객체의 자식 객체 중에서 특정 객체를 찾고자 할 때는 ( children(선택자) ) 메소드를 사용한다

### (2) 주어진 html 에서 제시된 문제를 해결하는 jQuery 코드를 작성하시오.

```
<body>
  <ul class="level-1">
    <li class="item-a">A
    <li class="item-b">B
      <ul class="level-2">
        <li class="item-1">1</li>
        <li class="item-2">2</li>
        <li class="item-3">3</li>
      </ul>
    </li>
    <li class="item-c">C</li>
  </ul>
</body>
```

- ① level-2 의 모든 자식 노드에 대하여 폰트 크기를 25px 로 한다

```
$('.level-2').children().css('font-size', '25px');
```

- ② level-2 의 첫번째 자식 노드만 선택하여 테두리를 표시한다. 굵기와 선모양, 색은 임의로 설정한다. eq() 사용.

```
$('.level-2').children().first().css('border', '2px solid blue');
```

- ③ item-2 의 모든 부모 노드에서 li 노드만 배경을 blue 로 한다

```
$('.item-2').parents('li').css('background-color', 'blue');
```

- ④ item-2 바로 다음 형제 노드를 선택하여 글자색과 폰트 크기를 임의의 값으로 설정한다.

```
$('.item-2').next().css({'color': 'blue', 'font-size': '30px'});
```

### (3) 형제 노드를 선택하기 위해 사용할 수 있는 jQuery 메소드를 제시하고 설명하시오

prev() - 이전 형제 객체를 선택한다.

prevAll(선택자) - 모든 이전 형제 객체 중 선택자에 해당하는 요소를 선택한다.

prevAll() - 이전 형제 객체를 모두 선택한다.

next() - 다음 형제 객체를 선택한다.

nextAll(선택자) - 모든 다음 형제 객체 중 선택자에 해당하는 요소를 선택한다.

nextAll() - 다음 형제 객체를 모두 선택한다.

(4) 부모 노드를 선택하기 위해 사용할 수 있는 jQuery 메소드를 제시하고 설명하시오

parent() – 부모 노드 요소를 선택한다.

parents(선택자) - 최상위 부모 노드 중에 해당되는 요소를 선택한다.

parents() - 최상위 부모 노드 요소 전체를 선택한다.

(5) 자식 노드를 선택하기 위해 사용할 수 있는 jQuery 메소드를 제시하고 설명하시오

children() – 자식 노드에 해당하는 요소를 선택한다.

children(선택자) – 자식 노드에 해당하는 요소 중 선택자에 해당하는 요소를 선택한다.

(6) 문서 노드를 추가하기 위해 사용할 수 있는 jQuery 메소드를 제시하고 설명하시오

\$(A).appendTo(B) – A 를 B 의 내부의 뒤 부분에 자식으로 추가한다

\$(A).append(B) – B 를 A 의 내부의 뒤 부분에 자식으로 추가한다

\$(A).prepend(B) – B 를 A 의 내부의 앞 부분에 자식으로 추가한다

\$(A).prependTo(B) – A 를 B 의 내부의 앞 부분에 자식으로 추가한다

\$(A).after(B) – B 를 A 의 뒤에 형제로 추가한다

\$(A).before(B) – B 를 A 의 앞에 형제로 추가한다

\$(A).insertAfter(B) – A 를 B 의 뒤에 형제로 추가한다

\$(A).insertBefore(B) – A 를 B 의 앞에 형제로 추가한다

\$(새요소).replaceAll(요소선택) – 선택한 요소를 새 요소로 교체한다.

\$(요소선택).replaceWith(새요소) – 선택한 요소를 새 요소로 교체한다.

(7) 다음 코드의 실행 결과를 제시하시오. 개발자 도구에서 확인하세요

//콘솔창으로 확인

```
<head>
<meta charset="UTF-8">
<style>
body {
    font-size: 9pt;
    font-family: "굴림";
}

ul.menu {
    padding: 10px;
    list-style: none;
    border: 1px #000 solid;
    width : 250px;
}

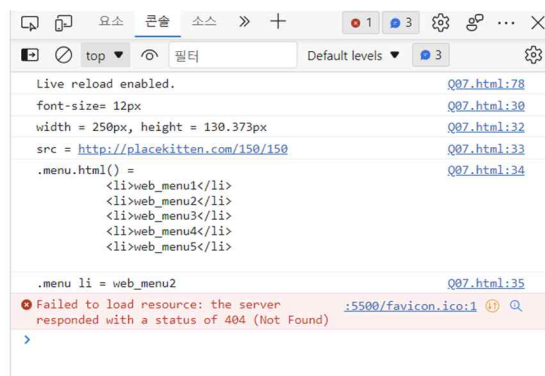
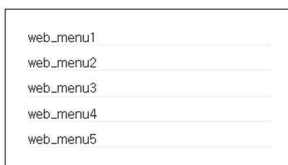
ul.menu li {
    border-bottom: 1px #eeeeee solid;
    margin: 10px;
}

li.select {
```

```

        background-color: #ccc;
    }
</style>
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function () {
        console.log(`font-size= ${$("ul.menu li").css("font-size")}`);
        const info = $("ul.menu").css(["width", "height"]);
        console.log(`width = ${info.width}, height = ${info.height}`);
        console.log(`src = ${$(".header img").attr("src")}`);
        console.log(`.menu.html() = ${$(".menu").html()}`);
        console.log(`.menu li = ${$(".menu").children().eq(1).text()}`);
    });
</script>
</head>
<body>
    <ul class="menu">
        <li>web_menu1 </li>
        <li>web_menu2</li>
        <li>web_menu3</li>
        <li>web_menu4</li>
        <li>web_menu5</li>
    </ul>
    <div class="header">
        
    </div>
</body>

```



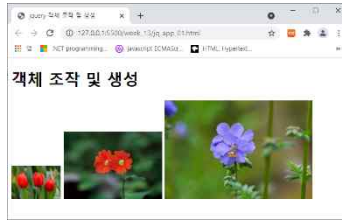
## □ 개념 활용 응용 프로그래밍

- (1) 다음과 같이 이미지를 출력하는 jQuery 코드를 작성하시오. 이미지와 크기는 본인이 임의로 지정하고 jQuery 속성 추가(attr) 메소드를 사용할 것.

```

<body>
  <h1>객체 조작 및 생성</h1>
  
  
  
</body>

```



## [소스]

```

<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function () {
    $('img').css('display', 'inline-block');
    $('img').first().css('height', '80px');
    $('img').eq(1).css('width', '160px');
    $('img').last().css('height', '240px');
  })
</script>

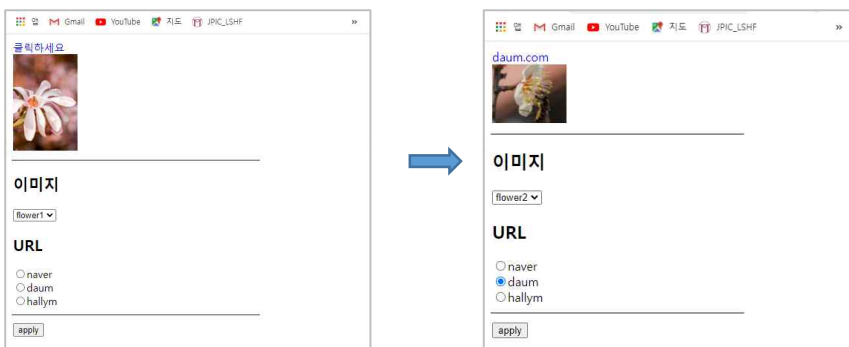
```

## [실행결과]

### 객체 조작 및 생성



- (2) 다음 문서에서 apply 버튼을 클릭하면 <img>태그의 src 속성값과 <a>태그의 href 속성값을 radio 버튼과 select 에서 선택한 값으로 변경하는 jQuery 코드를 작성하시오.
- 이미지와 url 을 제시된 html 에 더 추가하여 처리할 것 ☆
  - 스타일은 <style> 태그에 css 작성



```

<body>
  <div class="box1">
    <a href="#">클릭하세요</a>
    <br>
    

```

```
</div>
<hr>
<div class="box2">
  <h2>이미지</h2>
  <select id="imglist">
    <option value="imgs/flower1.jpeg">flower1 </option>
    <option value="imgs/flower2.png">flower2 </option>
    <option value="imgs/flower3.jpeg">flower3 </option>
  </select>
  <h2>URL</h2>
  <input type="radio" name="url" value="naver.com">naver<br>
  <input type="radio" name="url" value="daum.net">daum<br>
  <input type="radio" name="url" value="hallym.ac.kr">hallym
</div>
<hr>
<div class="choice">
  <button>apply</button>
</div>
</body>
```

---

#### [소스]

```
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function () {
    $('.box1 img').css('height', '210px');
    $('.choice button').click(function(){
      const imgval = $('option:selected').val();
      $('.box1 img').attr('src', imgval);


      const url = $('input:checked').val();
      const http = 'https://www.' + url;
      $('.box1 a').attr('href', http).text(url);
    });
  })
</script>
```

---

#### [실행결과]

---

클릭하세요



이미지


flower1

URL

☐ naver  
☐ daum  
☐ hallym

apply

daum.net



이미지

flower3

URL

☐ naver  
☒ daum  
☐ hallym

apply

(3) 추가와 삭제 버튼을 클릭하면 글 상자에 입력한 값과 일치하는 콘텐츠를 갖는 요소를 `<ul>` 태그에 추가하거나 삭제하는 jQuery 코드를 작성하시오

-추가 버튼 클릭 후 :

delete

추가

삭제

add

delete

-삭제 버튼 클릭 후:

delete

추가

삭제

add

```

<body>
  <div>
    <input type="text" id="menuName" />
    <button id="add">추가</button>
    <button id="remove">삭제</button>
  </div>
  <ul class="menu"> </ul>
</body>

```

[소스]

```

<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
  $(document).ready(function () {
    $('.menu').css('list-style', 'none');

    $('#add').click(function () {
      const name = $('#menuName').val();
      let item = "<li>" + name + "</li>";
      $('.menu').append(item);
      $('li:contains(' + name + ')').addClass(`${name}`).css({'border': '1px solid gray', 'margin': '10px'});
    });

    $('#remove').click(function () {
      const name = $('#menuName').val();

```

```

        $(`.${name}`).remove();
    });
})
</script>

```

### [실행결과]

- (4) 1초 간격으로 랜덤한 크기의 글자를 출력하는 jQuery 코드를 작성하시오. 단, 글자크기는 10px~40px
- 'play'버튼을 클릭하면 1초에 한번씩 <span> 태그를 신규 노드로 생성한 후 panel에 자식 노드로 추가한다. 글자 크기는 난수를 사용하여 처리한다.
  - 'stop' 버튼을 클릭하면 출력 작업을 중지한다
- 힌트 1) id=setInterval(콜백 함수, 시간) -> 시간 간격으로 콜백 함수 실행
- 힌트 2) clearInterval(id) -> 반복되어 실행되는 id 동작 중단
- 힌트 3) <span> 노드 생성 -> \$(`<span></span>`);
- 힌트 4) 자식 노드로 추가 -> append()

```

<body>
  <div>
    <button id="play">play</button>
    <button id="stop">stop</button>
  </div>
  <div id="panel">
    <h2>난수로 생성된 숫자 추가</h2>
  </div>
</body>

```



### [소스]

```

<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
  let cnt = 0;
  const text = function () {
    let ran = Math.random();
    let num = ++cnt;
    const spanum = $(`<span>${num}&nbsp;</span>`);
    $('#panel').append(spanum);

    let size = Math.floor(ran * 30 + 10);
    $(spanum).css('font-size', `${size}px`);

  };
  let play = null;
  $(document).ready(function () {
    $('#play').click(function () {
      play = setInterval(text, 1000);
    });
  });

```

```

    $('#stop').click(function () {
        clearInterval(play);
    });
})
</script>

```

#### [실행결과]

play stop

난수로 생성된 숫자 추가

1 2 3 4 5 6 7 8 9 10

- (5) 2 초 간격으로 박스 순서를 변경하는 jQuery 코드를 작성하시오. 변경 순서는 12345 -> 51234 -> 45123 와 같다. 단, 스타일은 \$(선택자).css(object); 방법을 사용하여 적용할 것

힌트) prependTo() 사용



```

<body>
<div>1</div>
<div>2</div>
<div>3</div>
<div>4</div>
<div>5</div>
</body>

```

#### [소스]

```

<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script>
    $(document).ready(function () {
        $('div').css({
            'width':'150px', 'height':'150px', 'display':'inline-block',
            'text-align':'center','line-height':'150px', 'margin':'10px'});
        $('div').first().css('background-color', 'red').addClass('first');
        $('div').eq(1).css('background-color', 'orange').addClass('two');
        $('div').eq(2).css('background-color', 'blue').addClass('three');
        $('div').eq(3).css('background-color', 'yellow').addClass('four');
        $('div').last().css('background-color', 'green').addClass('last');
        setInterval(function () {
            $('div').first().appendTo('body');
        },2000);
    })
</script>

```

#### [실행결과]

