

Javascript & jQuery

자바스크립트 & 제이쿼리 개념이해와 사용방법 익히기

- 자바스크립트 선언문, 제이쿼리 선언문, \$기호와 \$.noConflict() -
- 함수, 익명함수, 매개변수(Parameter), 아규먼트(Argument), IIFE(즉시실행함수표현식) -
- 변수, 증감변수, 선택자, alert(), console.log(), text(), html() 메소드 -
- ECMA SCRIPT 5 [변수선언 var]
- ECMA SCRIPT 6 [변수선언 let, const]
- 연산자, ++, --, 문자열 연결 연산자(+) -
- 이벤트핸들러, 이벤트, 객체지향 접근성 이벤트처리 -
- click, mouseenter, mouseleave, focusin, focusout, focus, blur -

```
<!DOCTYPE HTML> <!-- Document Object Modelling : DOM -->
<html>
<head>
  <title>Javascript & jQuery</title>
  <meta charset='utf-8'>
  <meta name='Publisher' content='교사문선종'>

  <style>
    body { text-align:center; padding-top:100px; }
    button { width:200px; background:#000; color:#fff; height:60px; font-size:25px; }
    button:focus { background:#c00; }

    h1 { padding-top:50px; }
  </style>
```

```
<!-- 제이쿼리 라이브러리 프로그램 연동 -->
<script src='./js/lib/jquery-1.12.4.min.js'></script>
<script>
    //자바스크립트 script 내에서 곧바로 사용 가능
    //alert('자바스크립트 경고창!!!'); //알럿트 경고창 메소드('메시지내용');
    //alert('자바스크립트 경고창!!! \n프론트엔드개발자 교사 문선종'); //알럿트 경고창 메소드('메시지내용');
    //alert('자바스크립트 경고창!!! \n    프론트엔드개발자 교사 문선종'); //알럿트 경고창 메소드('메시지내용');
```

```
/*
    //Document Object Modelling : DOM 로딩 ready 준비되면 제이쿼리 사용가능
    $(document).ready(function(){
        console.log('제이쿼리 로딩시 웹문서 모두 읽어들이고 준비되면 제이쿼리 실행');
    });

    $(function(){ //제이쿼리 선언문
        alert('제이쿼리실행1 테스트');
    });

    jQuery(function(){ //제이쿼리 선언문
        alert('제이쿼리실행2 테스트');
    });

    //$ == jQuery 제이쿼리 표현식 기호
    //다른 플러그인에서 $기호가 충돌 가능성
    //$ 사용안하겠다 noConflict();
    $.noConflict();
    jQuery(function(){
        alert('$사용안함!!!!');
    });
```

```
//ECMA SCRIPT 5버전 변수선언시 앞에 접두어가 var 변수이름 선언  
//ECMA SCRIPT 6버전 변수선언시 앞에 접두어가 let const 변수이름 선언  
//$대신 다른문자나 기호를 사용가능  
var jq = $.noConflict();  
    jq(function(){  
        alert('jq 문자로 $ 대신 사용함.');    });
```

```
//보호막을 만들어서 $ 사용을 변수로 가능하게 만든다.  
//함수식을 만들어서 아규먼트를 매개변수에 전달하여 사용가능 하다.  
//함수 표현식 IIFE(즉시실행함수) 개발자 개발 함수식 사용
```

```
(function(매개변수1,매개변수2,매개변수3){ parameter(파라미터)  
    alert('실행문');  
})(아규먼트1,아규먼트2,아규먼트3); agument(아규먼트)
```

```
(function($, window, document, undefinde){  
    alert('실행문');  
})(jQuery, window, document);
```

```
//즉시실행함수에서 보호막을 () 이 함수표현식 안에서 아규먼트로 jQuery를 매개변수 $ 기호 사용  
(function($){  
  
    alert('제이쿼리 jQuery사용변수 $기호사용');  
    console.log( $ );  
  
})(jQuery);
```

```
//함수는 명령어 또는 프로그램등이 1개이상 묶어서 실행 하는 그룹단위
```

```
//이름있는 함수
함수이름(); //실행
function 함수이름(){
    명령어 (메소드) 실행문
}
함수이름(); //실행
```

```
////////////////////////////////////
//이름없는 함수(익명함수)
변수이름(); //함수 선언문 위에서는 실행 불가
var 변수이름 = function(){ //익명함수

}
변수이름();
```

//카멜표기법 : 함수이름, 변수이름에 두단어함수시 두번째단어 첫글자 대문자표현형식

```
//var a=-1; //증가변수 초기값 -1 ? 1씩증가 처음 결과값 0부터 사용하기 위해서
var a=0; //증가변수 초기값 0 ? 1씩증가 처음 결과값 1부터 사용하기 위해서
```

```
slideFn(); //함수호출 실행1
function slideFn(){
    a++;
    alert(a + '이름있는 함수 실행내용!!!');
    console.log(a + '콘솔출력 이름있는 함수 실행내용!!!');
}
slideFn(); //함수호출 실행2
slideFn(); //함수호출 실행3
```

```
//익명함수
```

```
var b=-1;
```

```
var slideFn = function(){  
    b++; //0 1 2 3  
    console.log('익명함수 호출 실행' + b + '번');  
}
```

```
slideFn(); //호출실행 1  
slideFn(); //호출실행 2  
slideFn(); //호출실행 3  
slideFn(); //호출실행 4
```

```
//즉시실행함수 표현식 (IIFE) 이피
```

```
(function(a,b,c,d){  
    console.log(a);  
    console.log(b);  
    console.log(c);  
    console.log(d);  
})(jQuery,window,document);
```

```
;(function($,window,document,undefined){
```

```
    console.log($);  
    console.log(window);  
    console.log(document);  
    console.log(undefined);
```

```
})(jQuery,window,document);  
*/
```

```
</script>
```

```

</head>
<body>

    <!--
        버튼(Btn1)을 클릭(click)하면 클릭할 때 마다 1씩(증가변수 cnt) 증가
        그리고 증가한 숫자를 텍스트로(outText) 출력
    -->
    <button class='Btn1'> 버튼 </button>
    <h1 class='outText'>출력내용</h1>

    <script>
        //제이쿼리 라이브러리에서 버튼클릭 이벤트 처리
        //제이쿼리 정의문(선언문)
        //즉시실행함수 표현식 익명함수 반드시 HTML,CSS 모두 로딩하고 다음 실행

        //1. HTML 마크업 요소 로딩          - Modelling : M
        //2. CSS 스타일 속성 로딩           - View      : V
        //3. 애니메이션,이벤트....등처리    - Controler : C
        //  MVC 모델링

        ;(function($){
            var cnt = 0;
            /*
            $('Btn1').click(function(){
                cnt++;
                //$('.outText').text( cnt ); //text()메소드 : 태그사용 불가능
                $('.outText').html('<i style="font-size:100px"><u>' + cnt + '</u></i>'); //html()메소드 : 태그사용 가능
                //$('.outText').text('<i style="font-size:100px"><u>' + cnt + '</u></i>'); //html()메소드 : 태그사용 가능
            });

            $('Btn1').on('click', function(){

```

```
        cnt++; //1 2 3 4 5 ....
        $('#outText').text(cnt);
    });
```

```
$('#Btn1').on({
    click: function(){ //엔터 == 클릭
        cnt++;
        $('#outText').text(cnt);
    }
});
```

*/

//객체지향 이벤트방식

//접근성 고려한 테스트 이벤트 내용

```
$('#Btn1').on({
    click: function(){ //엔터
        cnt++;
        $('#outText').text(cnt);
    },
    mouseenter: function(){ //마우스 접근하면(오버) mouseover
        cnt++;
        $('#outText').text(cnt);
    },
    mouseleave: function(){ //마우스 떠나면(아웃) mouseout
        cnt--;
        $('#outText').text(cnt);
    },
    focusin: function(){ //키보드가 접근하면 focus
        cnt++;
        $('#outText').text(cnt);
    },
});
```

```
focusout: function(){ //키보드가 떠나면 blur
    cnt--;
    $('.outText').text(cnt);
}
```

```
});
```

```
})(jQuery);
```

```
</script>
```

```
</body>
```

```
</html>
```