

2강 포트폴리오 제작을 위한 이벤트 인터랙션1 둘째 강좌 : Event Interaction

0201.html :

```
<script>
var moving=false;
var num=0;

function doLayout(){
    // console.log("doLayout function");
    var w=window.innerWidth*(-1)*num;
    // console.log("w : "+w);

    $(".section").css({left:w});
    $(".section .page").removeClass("loading")
    .each(function(){
        // console.log("index : "+$(this).index());
        $(this).css({left:window.innerWidth*$(this).index(), width:window.innerWidth,
height:innerHeight});
    });
}

function controlBtn(){
    // console.log("controlBtn function");
    $(".aside .item").removeClass("on");
    $(".aside .item").eq(num).addClass("on");

    if(num == 0){
        $(".prev").removeClass("on");
        $(".next").addClass("on");
    }else if(num == 4){
        $(".prev").addClass("on");
        $(".next").removeClass("on");
    }else{
        $(".prev").addClass("on");
        $(".next").addClass("on");
    }
}

$(function(){
    doLayout();
    controlBtn();

    // 1) aside click event
    $(".aside .item a").click(function(e){
```

```

e.preventDefault();
var order=$(this).parent("li").index();
num=order;
// console.log("order : "+order);
var offsetL=window.innerWidth*(-1)*num;
$(".aside .item").removeClass("on");
$(".aside .item").eq(num).addClass("on");

if(!moving){
    moving=true;
    $(".section").animate({left:offsetL}, 800, function(){
        moving=false;
        controlBtn();
    });
}
});

// 2) next click event
$(".next").click(function(){
    var w=window.innerWidth;
    var offsetL=$(".section").offset().left;

    if(!moving){
        moving=true;
        if(num < 4){
            num+=1;
            // console.log("num : "+num);
            offsetL-=w;
            $(".section").animate({left:offsetL}, 800, function(){
                moving=false;
                controlBtn();
            });
        }
    }
});

// 3) prev click event
$(".prev").click(function(){
    var w=window.innerWidth;
    var offsetL=$(".section").offset().left;

    if(!moving){
        moving=true;
        if(num > 0){

```

```

        num-=1;
        // console.log("num : "+num);
        offsetL+=w;
        $(".section").animate({left:offsetL}, 800, function(){
            moving=false;
            controlBtn();
        });
    }
}
});

```

// 4) keybord event

```

$(this).keyup(function(e){
    // console.log("keybord event");
    // console.log("e.keyCode : "+e.keyCode);
    if(!moving){
        moving=true;
        var w=window.innerWidth;
        var offsetL=$(".section").offset().left;

        if(e.keyCode == 37){
            if(num < 4){
                num+=1;
                offsetL-=w;
            }
        }else if(e.keyCode == 39){
            if(num > 0){
                num-=1;
                offsetL+=w;
            }
        }
        $(".section").animate({left:offsetL}, 800, function(){
            moving=false;
            controlBtn();
        });
    }
});

```

// 5) mouse wheel event

```

$(".section").mousewheel(function(e){
    if(!moving){
        moving=true;
        var w=window.innerWidth;
        var offsetL=$(".section").offset().left;
    }
});

```

```

        // console.log(e.deltaX, e.deltaY, e.deltaFactor);

        if(e.deltaY == -1){
            if(num < 4){
                num+=1;
                offsetL-=w;
            }
        }else if(e.deltaY == 1){
            if(num > 0){
                num-=1;
                offsetL+=w;
            }
        }
        $(".section").animate({left:offsetL}, 800, function(){
            moving=false;
            controlBtn();
        });
    }
});
</script>

```

이번 강좌에서 만들 상호작용 (Interaction)입니다. :

- 클릭 이벤트
- 키보드 이벤트
- 마우스 휠 이벤트

line039 : var moving=false;

현재 움직이는 상황 여부를 점검하는 변수입니다.

line040 : var num=0;

키비주얼의 위치 번호를 나타내는 변수입니다.

line044 : var w=window.innerWidth*(-1)*num;

window.innerWidth : 윈도우 창의 가로 크기 영역입니다.

window.innerWidth*(-1)*num : 메인 키비주얼 (.section)의 가로 위치입니다.

line048 : \$(".section .page").removeClass("loading")

loading 클래스를 제거하면 내부 영역 (.page)이 보이게 됩니다.

line051 : \$(this).css({left:window.innerWidth*(this.index()), width>window.innerWidth,

```
height:innerHeight));
```

left : 윈도우 창의 가로 크기 만큼 순서대로 배치됩니다. (0, 1, 2, 3, 4)

width : 윈도우 창의 가로 크기입니다.

height: 윈도우 창의 세로 크기입니다.

```
line056 : $(".aside .item").removeClass("on");
```

활성화된 버튼을 초기화합니다.

```
line057 : $(".aside .item").eq(num).addClass("on");
```

현재 위치 번호에 맞는 선택 버튼만 활성화됩니다.

```
line059 : if(num == 0){
```

num 번호에 맞게 prev, next 클래스가 보이고 안보이는 여부가 정해집니다.

```
line060 : $(".prev").removeClass("on");
```

적용된 CSS는 다음과 같습니다.

```
.prev.on {  
    left: 0;  
}
```

```
line061 : $(".next").addClass("on");
```

적용된 CSS는 다음과 같습니다.

```
.next.on {  
    right: 0;  
}
```

```
line065 : }else{
```

이외의 경우는 1, 2, 3의 경우입니다.

```
line071 : doLayout();
```

레이아웃을 초기화하는 함수를 호출합니다.

```
line072 : controlBtn();
```

버튼을 초기화하는 함수를 호출합니다.

```
line076 : e.preventDefault();
```

URL 이동이 아닌 기능 적용을 위한 구조를 만들 때 필수적인 코드입니다.

```
line077 : var order=$(this).parent("li").index();
```

현재 클릭된 번호를 정해줍니다.

```
line078 : num=order;
```

키비주얼의 위치 번호를 order 변수로 정해줍니다.

```
line084 : if(!moving){
```

현재 애니메이션이 일어나지 않는 상황입니다.

```
line086 : $("section").animate({left:offsetL}, 800, function(){
```

animate 메서드의 용법은 아래와 같습니다.

```
animate({property:value}, second, [complete function]);
```

```
line095 : var w=window.innerWidth;
```

윈도우 창의 가로 크기입니다.

```
line096 : var offsetL=$("section").offset().left;
```

section 클래스의 가로 위치입니다.

```
line103 : offsetL-=w;
```

좌표가 윈도우 창의 가로 크기만큼 왼쪽으로 이동할 좌표를 준비합니다.

```
line122 : offsetL+=w;
```

좌표가 윈도우 창의 가로 크기만큼 오른쪽으로 이동할 좌표를 준비합니다.

```
line134 : // console.log("e.keyCode : "+e.keyCode);
```

37은 왼쪽 버튼이고, 39는 우측 버튼이 됩니다.

왼쪽 버튼은 next 클래스 버튼을 누른 내용이 적용되고, 오른쪽 버튼은 prev 클래스 버튼이 누른 내용이 적용이 됩니다.

```
line136 : moving=true;
```

반복적인 이벤트를 방지하기 위해 적용합니다. 키보드 이벤트와 마우스 휠 이벤트에서는 필수적으로 필요한 변수입니다.

```
line140 : if(e.keyCode == 37){
```

왼쪽 버튼입니다.

```
line145 : }else if(e.keyCode == 39){
```

오른쪽 버튼입니다.

```
line164 : // console.log(e.deltaX, e.deltaY, e.deltaFactor);
```

e.deltaY는 위로 휠하면 1, 아래로 휠하면 -1입니다.

e.deltaY == 1일 경우에는 prev 클래스 버튼을 누른 내용이 적용됩니다.

e.deltaY == -1일 경우에는 next 클래스 버튼을 누른 내용이 적용됩니다.

```
line166 : if(e.deltaY == -1){
```

아래로 휠 할 경우가 됩니다.

```
line171 : }else if(e.deltaY == 1){
```

위로 휠 할 경우가 됩니다.