

14강 jQuery와 Media Queries를 활용한 Responsive Hamburger Toggle Menu

style.css :

```
@charset "UTF-8";

body {
    margin: 0;
    padding: 0;
    font-family: "Open Sans", sans-serif;
    font-size: 1em;
    background: #f1f1f1;
}

.container {
    margin-left: 0;
    -webkit-transition: margin-left 0.5s ease-out;
    -moz-transition: margin-left 0.5s ease-out;
    transition: margin-left 0.5s ease-out;
}

/* header */
header {
    display: block;
    position: fixed;
    top: 0;
    z-index: 100;
    width: 0;
    overflow: hidden;
    background: #fff;
    -webkit-box-shadow: 1px 1px 5px rgba(0,0,0,.1);
    -moz-box-shadow: 1px 1px 5px rgba(0,0,0,.1);
    box-shadow: 1px 1px 5px rgba(0,0,0,.1);
}

.menu_btn {
    display: none;
    position: relative;
    z-index: 101;
    top: 0;
    margin: 28px 0 0 20px;
    width: 40px;
    height: 40px;
    line-height: 0;
    text-decoration: none;
```

```
    text-indent: -9999px;
    background: url(../images/toggle_button.png) no-repeat 0 0;
}
nav {
    margin: 0 auto;
}
nav .hamburger_menu {
    position: fixed;
    z-index: 101;
    left: 0;
    top: 0;
    width: 250px;
    height: 100%;
    overflow: hidden;
    background: #fff;
    -webkit-box-shadow: 1px 1px 5px rgba(0,0,0,.1);
    -moz-box-shadow: 1px 1px 5px rgba(0,0,0,.1);
    box-shadow: 1px 1px 5px rgba(0,0,0,.1);
    -webkit-transition: left 0.5s ease-out;
    -moz-transition: left 0.5s ease-out;
    transition: left 0.5s ease-out;
}
nav .hamburger_menu .logo {
    margin-top: 100px;
    margin-left: 50px;
}
nav .hamburger_menu .logo a {
    text-decoration: none;
    font-family: "Open Sans", sans-serif;
    font-size: 1.125em;
    font-weight: 600;
    color: #555;
}
nav .hamburger_menu ul {
    z-index: 101;
    margin-left: 50px;
    margin-top: 50px;
}
nav .hamburger_menu ul li {
    display: block;
    margin-bottom: 20px;
}
nav .hamburger_menu ul li a {
    text-decoration: none;
```

```

        font-size: 1em;
        color: #555;
    }
    nav .hamburger_menu li a:hover {
        text-decoration: underline;
    }
    .overlay {
        position: fixed;
        z-index: 100;
        left: 0;
        right: 0;
        bottom: 0;
        background: rgba(0,0,0,.5);
        -webkit-transition: left 0.5s ease-out;
        -moz-transition: left 0.5s ease-out;
        transition: left 0.5s ease-out;
    }
    .overlay.on {
        left: 250px;
    }
    .article {
        margin-left: 250px;
        padding: 160px 50px 0;
        min-width: 500px;
    }
    /* 하위 요소는 스크립트 적용 후에 설명을 드리겠습니다. */
    .open_status .container {
        margin-left: 250px;
    }
    .open_status .overlay {
        top: 0;
    }
    .open_status .menu_btn {
        background-position: -41px 0;
    }

    /* responsive CSS */
    @media (max-width: 750px) {
        header {
            display: block;
            left: 0;
            position: relative;
            padding: 0;
            width: 100%;

```

```

        min-height: 100px;
    }
    .article {
        margin-left: 0;
        padding: 30px;
    }
    .menu_btn {
        display: inline-block;
    }
    nav .hamburger_menu {
        left: -250px;
    }
    nav .hamburger_menu.on {
        left: 0;
    }
}

```

line011 : margin-left: 0;

초기의 margin-left 속성 값을 지정합니다.

line020 : position: fixed;

height 속성을 100%로 만들 수 있습니다.

line023 : width: 0;

width 속성 값이 0이므로 보이지 않습니다. width: 100%;로 테스트합니다.

line030 : .menu_btn {

Hamburger 메뉴 영역입니다. 이 부분은 모바일 해상도에서 노출됩니다.

line031 : display: none;

display: block으로 테스트합니다.

line032 : position: relative;

relative 속성은 absolute 방식 (left, right, top, bottom)으로 정렬이 되며, static 방식 (margin, padding)으로도 정렬됩니다.

line040 : text-indent: -9999px;

해당 텍스트는 안보이도록 처리합니다.

line104 : .article {

실제 콘텐츠가 배치될 영역입니다.

line110 : .open_status .container {

전체 영역이 이동됩니다.

line113 : .open_status .overlay {
overlay가 화면에 보이게 됩니다.

line116 : .open_status .menu_btn {
hamburger 모양의 아이콘이 바뀌게 됩니다.

line121 : @media (max-width: 750px) {
750px 이하의 해상도에 해당됩니다.

line125 : position: relative;
position: fixed; 속성에서 변경됩니다.

line127 : width: 100%;
width: 0; 속성에서 변경됩니다.

line128 : min-height: 100px;
해상도가 변경되면 추가되는 새로운 속성입니다.

line135 : display: inline-block;
display: none; 속성에서 변경됩니다.

line138 : left: -250px;
left: 0; 속성에서 변경됩니다.

main.js :

```
$(function(){
    var $body=$("body");
    var $container=$(".container");
    var $btn=$(".menu_btn");
    var $menu=$(".hamburger_menu");
    var $overlay=$(".overlay");

    $("a[href$='#']").on("click", function(e){
        e.preventDefault();
    });
    $btn.on("click", function(){
        if($body.hasClass("open_status")){
            toggleMenu("close");
        }else{
            toggleMenu("open");
        }
    });
    $overlay.on("click", function(){
        toggleMenu("close");
    });
    function toggleMenu(str){
        if(str == "close"){
            $body.removeClass("open_status");
            $menu.removeClass("on");
            $overlay.removeClass("on");
        }else if(str == "open"){
            $body.addClass("open_status");
            $menu.addClass("on");
            $overlay.addClass("on");
        }
    }
});
```

line001 : \$(function){
\$(document).ready(function){

line002 : var \$body=\$("body");
나름대로의 참조 요소에 대한 규칙을 만듭니다.

line008 : \$("a[href\$='#']").on("click", function(e){
#을 가진 null 링크 요소를 클릭했을 때의 이벤트입니다.

line009 : e.preventDefault();

링크 처리를 하지 않습니다. (return false;)

line012 : if(\$body.hasClass("open_status")){

메뉴가 열린 경우에 닫아주는 함수를 호출합니다. (close)

line014 : }else{

메뉴가 닫힌 경우에 열어주는 함수를 호출합니다. (open)

line019 : toggleMenu("close");

메뉴가 열린 경우에 닫아주는 함수를 호출합니다. (close)

line022 : if(str == "close"){

메뉴가 열린 경우에 닫아주는 영역을 작성합니다.

line026 : }else if(str == "open"){

메뉴가 닫힌 경우에 열어주는 영역을 작성합니다.

line027 : \$body.addClass("open_status");

현재 열린 메뉴를 확인하는 클래스입니다. (open_status 클래스)

line028 : \$menu.addClass("on");

on 클래스가 적용이 된다면 좌표가 이동됩니다. (-250px ---> 0)

line029 : \$overlay.addClass("on");

on 클래스가 적용이 된다면 좌표가 이동됩니다. (0 ---> 250px)