

웹프로그래밍 기초

---

# 개발자 환경 구축

모바일웹 UI/UX 디자인 & 프론트엔드(React.js) B

---

평가 일시 2022년 12월 12일(월)

---

평가 방법 문제해결시나리오, 서술형시험

---

훈 련 생 조소영 (인)

---

2022. 12. 12.

# 개발자 환경구축

Z Scroll(gsap)

조소영



# Contents

## Part1 Z축 스크롤 웹페이지제작에 대한 기획 및 스토리보드 작성

01 웹페이지 콘셉트

02 UI정보와 레이아웃 설명

03 스토리보드

## Part 2 결과물 코드 리뷰

PART 1

# **Z축 스크롤 웹페이지제작에 대한 기획 및 스토리보드 작성**

# 01 웹페이지 콘셉트

## Theme

바다 생물, 고래종류에 대한 소개

## Font

*Playball*

*A dolphin is an aquatic mammal within the infraorder Cetacea.*

## Color

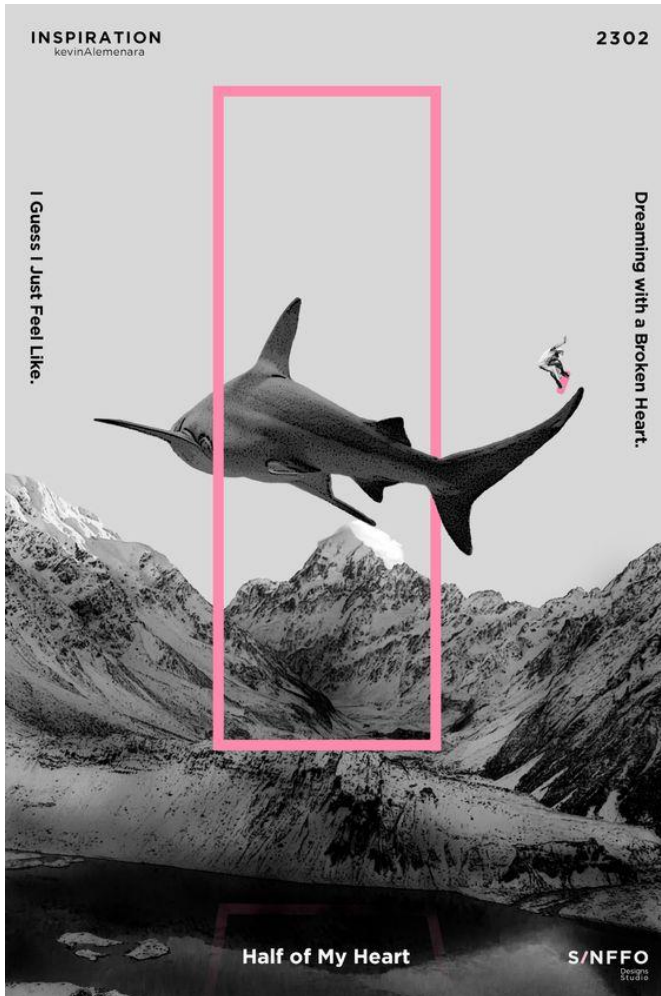
#aeaead

#ffffff

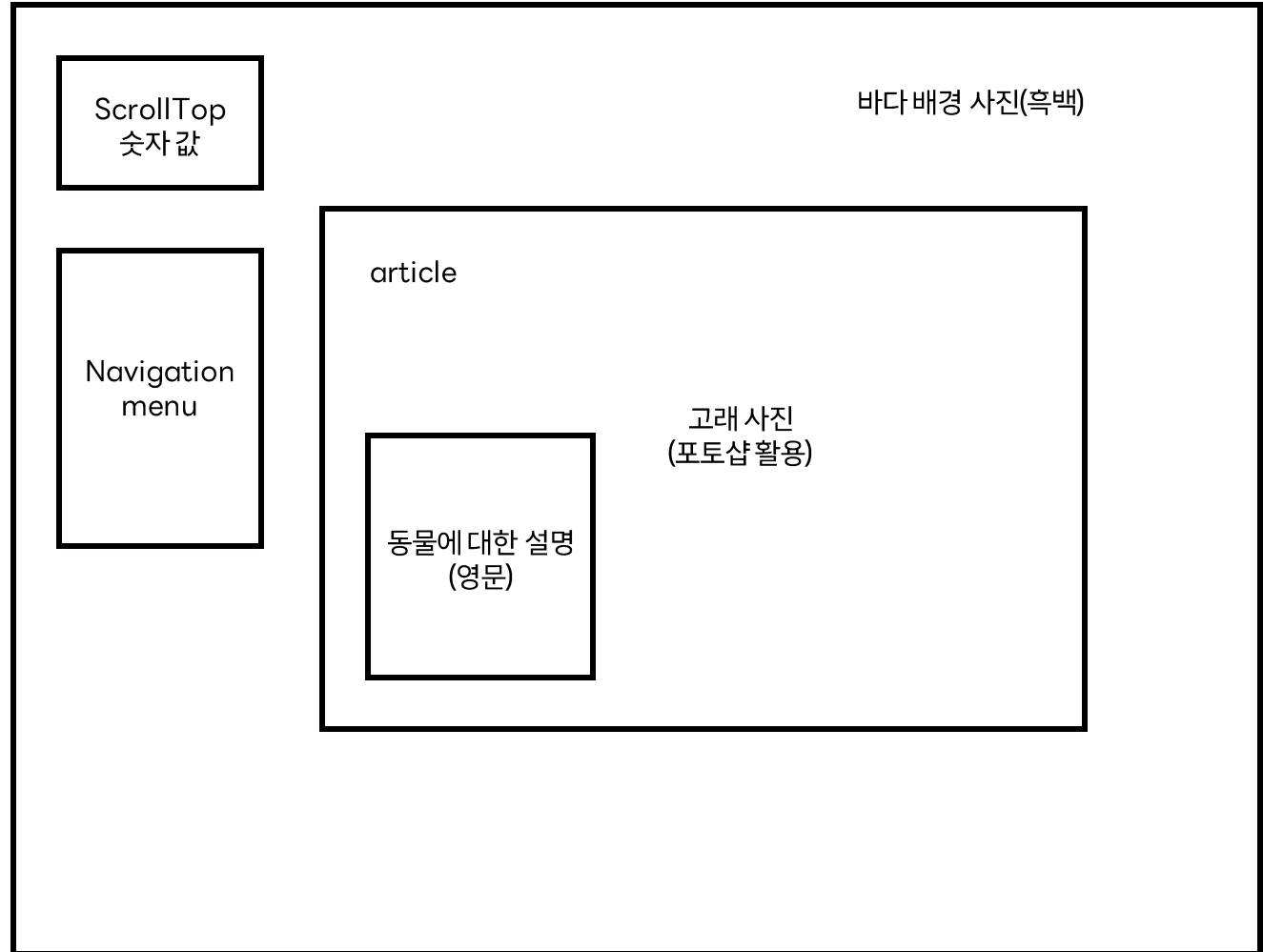
#000000

## 02 UI정보와 레이아웃 설명

### Reference



### Layout





## 03 스토리보드

Scroll을 할 때마다 scrollTop 메소드로 받은 스크롤 값이 왼쪽 상단에 숫자로 표시 된다.



스크롤을 내릴 때마다 해당 섹션에 해당하는  
내비게이션의 폰트 색깔이 검은색에서 흰색으로 바뀐다.  
버튼을 누르면 해당 섹션으로 이동한다.

PART 2

# 결과물 코드 리뷰



## transform(3D변형)

```
section > article:nth-child(3){
  transform: translateZ(-10000px);
}
section > article:nth-child(3) h2{
  top:40%; right:20%;
}

section > article:nth-child(3) p{
  bottom:15%; right:15%;
}

section > article:nth-child(4){
  transform: translateZ(-15000px);
}
```

## 이벤트 연결(클릭시 해당 섹션으로 스크롤하여 이동)

```
//스크롤 내비게이션 클릭시 스크롤 이동.
$(".scrollNavi li").on("click", function(){
  var a = $(this).index();
  $("body,html").stop().animate({"scrollTop": 5000*a},1500,"swing");
});
```

.scrollTop(): 현재 스크롤된 거리만큼 저장하거나 실제 해당 위치로 문서를 스크롤 시킴.

```
$(function(){
    // 페이지 로딩시 스크롤이 상단으로 이동. h1에 scrollTop의 위치값이 표기.
    $("body, html").stop().animate({"scrollTop":0}, 1500,"swing");
    //변수 scroll에 현재 scrollTop의 위치값을 저장하고 h1 출력
    $(window).on("scroll", function(){
        var scroll = $(this).scrollTop();
        $("h1").text(scroll);
        //스크롤 값과 박스의 z축을 연동하고 확장구간에 스크롤 내비게이션 텍스트를 활성화
        /*
        parseInt()함수는 문자열을 숫자열로 반환.
        1. 변수i의 초기값은 0이며 i가 5보다 작을 때 다음구문 실행.
        2. 각 article의 인덱스는 z축으로 -5000 * i + scroll(현재 스크롤의 위치값)으로 이동하고 다음 조건식 실행.
        3. scroll(현재 스크롤 위치값.)이 (해당인덱스 * 5000)보다 크거나 작고, (해당 인덱스 +1) * 5000보다 적으면 true를 반환하여
        다음 {}구문 실행. */
        for(var i=0; i<5; i++){
            $("section>article").eq(i).css({"transform": "translateZ(" + parseInt((-5000*i)+scroll) + "px)"});
            if(scroll >= i*5000 && scroll < (i+1)*5000){
                $(".scrollNavi li").removeClass('on');
                $(".scrollNavi li").eq(i).addClass('on');
                $("article").removeClass('on');
                $("article").eq(i).addClass('on');
            }
        }
    });
});
```

if 조건문과 for 반복문 사용

.addClass(), .removeClass()

스크롤 해서 특정한 z축에 들어올 때마다 해당 영역에 on 클래스를 붙이고, 다른 영역들의 on 클래스는 제거함.

```
for(var i=0; i<5; i++){
    $("section>article").eq(i).css({"transform": "translateZ("+ parseInt((-5000*i)+scroll) +"px)"});
    if(scroll >= i*5000 && scroll < (i+1)*5000){
        $(".scrollNavi li").removeClass('on');
        $(".scrollNavi li").eq(i).addClass('on');
        $("article").removeClass('on');
        $("article").eq(i).addClass('on');
    };
};
```

mousemove() 이벤트 적용: 마우스 움직임에 따라 콘텐츠가 움직임

```
//05. 화면에서 마우스 무브시 박스안의 콘텐츠 움직이기
$("body").on("mousemove", function(e){
    var posX = e.pageX/50; //변수 posX에는 현재 화면상의 마우스 커서의 x축의 위치
    var posY = e.pageY/70; //변수 posY에는 현재 화면상의 마우스 커서의 Y축의 위치.
    //이때 해당수치의 변화폭을 줄이기 위해 수치를 나눠준다.(50~150)
    $("article").eq(0).css({"left":posX, "top":posY});
    $("article").eq(1).css({"left":posX, "top":posY});
    $("article").eq(2).css({"left":posX-10, "top":posY-50});
    $("article").eq(3).css({"left":posX+30, "top":posY-200});
    $("article").eq(4).css({"left":posX-30, "top":posY-250});
});
});
});
```