

NCS능력단위평가 결과물제출

능력단위명	REACTJS
성 명	이채윤
제출일자	2021-12-27

문항1

이채윤

01. 기능구현 페이지

기능구현 페이지 명 : 메인페이지 view영역 가로형 광고슬라이드 기능

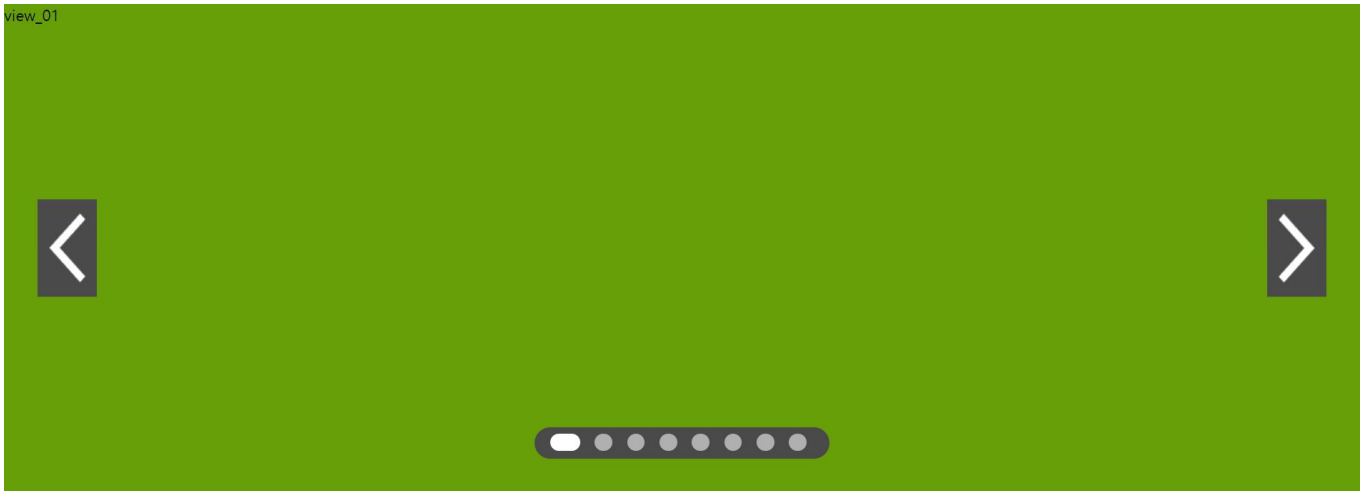
설명

버튼 클릭시 해당 순번에 맞는 광고 화면 표시 - 스크롤 이동(laptop, pc)

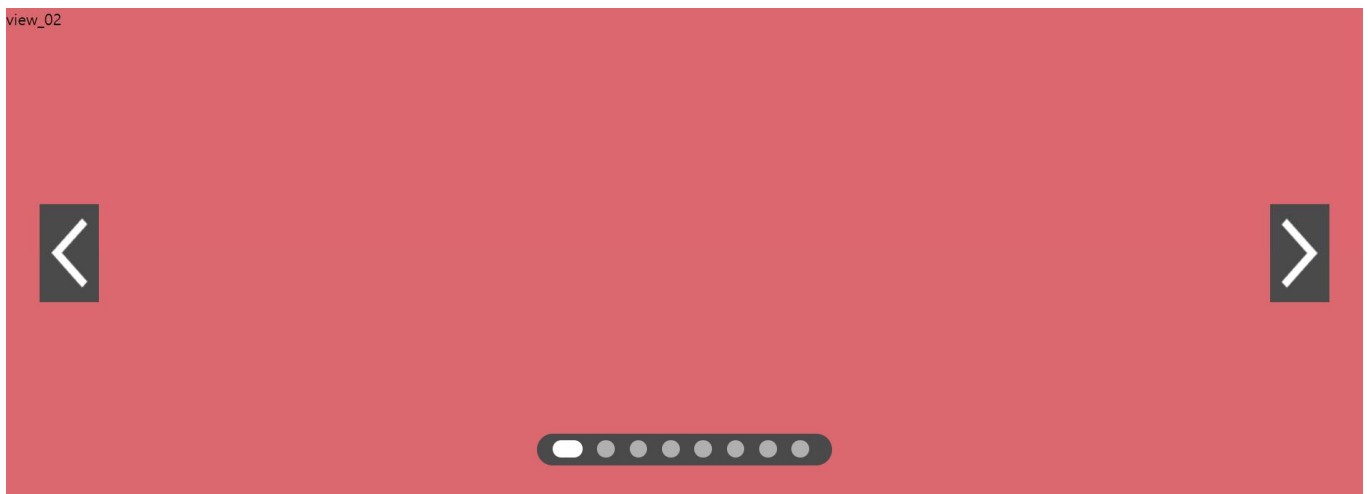
* 인디케이터 제외

기능구현 이미지

view_01



view_02



02. 기능구현 페이지 관련 코드

HTML

```
<section id="viewBox">
  <h2 class="blind">오투기 주요 소식</h2>
  <div class="view_btn_area">
    <button type="button" class="next"><span>next</span></button>
    <button type="button" class="prev"><span>prev</span></button>
  </div>
  <div class="indicator"> ...
</div>
<ul class="view_area">
  <li id="view_01">
    <p>view_01</p>
    <!-- <a href="#">자세히보기</a> -->
  </li>
  <li id="view_02"> ...
</li>
  <li id="view_03"> ...
</li>
  <li id="view_04"> ...
</li>
  <li id="view_05"> ...
</li>
  <li id="view_06"> ...
</li>
  <li id="view_07"> ...
</li>
  <li id="view_08"> ...
</li>
</ul>
</section><!-- #viewBox -->
```

02. 기능구현 페이지 관련 코드

CSS

```
#viewBox {
  position: relative; width: 100%; overflow: hidden;
  // @include img('view_bg.png', $size: cover); // 임시
  @include mobile { @include remUnit(height, 300); }
  @include tablet { @include remUnit(height, 360); }
  @include laptop { @include remUnit(height, 500); }
  @include pc { @include remUnit(height, 500); }
} // #viewBox

.view_btn_area {
  position: absolute; top: 50%; left: 0; right: 0; z-index: 5;
  width: 95%; height: auto;
  margin: auto; transform: translateY(-50%);
  button { width: 60px; height: 100px; background-color: $color_gray_04; }
  .next { float: right; @include img('view_next.png'); }
  .prev { float: left; @include img('view_prev.png'); }
  @include mobile { display: none; }
  @include tablet { display: none; }
}

.indicator { --
}

.view_area {
  display: flex; flex-wrap: nowrap;
  width: 800%; height: 100%;
  > li {
    position: relative; z-index: 0; width: calc(100% / 8); height: 100%; }
  // a { display: block; width: 100%; height: 100%; }
  #view_01 { background-color: #66bb6a; }
  #view_02 { background-color: #e57373; }
  #view_03 { background-color: #9575cd; }
  #view_04 { background-color: #00bcd4; }
  #view_05 { background-color: #ffeb3b; }
  #view_06 { background-color: #9c27b0; }
  #view_07 { background-color: #00bcd4; }
  #view_08 { background-color: #ff0000; }
}
```

문항1

이채운

02. 기능구현 페이지 관련 코드

JS - 1

```
// 변수 -----

const elViewBox = document.querySelector('#viewBox');
const slideNext = elViewBox.querySelector('.next');
const slidePrev = elViewBox.querySelector('.prev');

const viewArea = elViewBox.querySelector('.view_area');
const elSlideLi = viewArea.querySelectorAll('li');
const elSlideCvt = [].slice.call(elSlideLi);

let SLIDE_COUNT = 0;
let TIME_OPTION = 500;
let PERMISSION = true;

const slideLen = elSlideLi.length;

// 슬라이드 기능 -----

// 1. 마지막 요소를 복제하여 첫번째 요소 앞에 위치시킨다
const elSlideLast = elSlideCvt.at(-1);
const cloneSlide = elSlideLast.cloneNode(true);
viewArea.prepend(cloneSlide);

// 2. ul의 크기를 증가시킨다(800% -> 900%)
const ulStyle = viewArea.style;
ulStyle.width = ((slideLen + 1) * 100) + '%';

// 3. li의 크기를 감소시킨다(100%/8 -> 100%/9)
const elSlideLiRe = viewArea.querySelectorAll('li');
const elSlideLiReCvt = [].slice.call(elSlideLiRe);

elSlideLiReCvt.forEach((li, idx) => {
  li.style.width = 100 / (slideLen + 1) + '%';
});

// * 첫번째 요소가 보이게 1칸 이동
ulStyle.position = 'relative';
ulStyle.left = 0;
ulStyle.marginLeft = '-100%';

ulStyle.transition = `left ${TIME_OPTION}ms linear`;
```


02. 기능구현 페이지 관련 코드

JS - 2

```
// 함수 -----
// setTimeout 간략화 함수
const fnDelay = async (ms = 0) => {
  return await new Promise(resolve => {
    setTimeout(() => { resolve() }, ms)
  });
}

// 다음버튼 함수
const fnNextAni = async () => {
  if(SLIDE_COUNT >= slideLen){
    SLIDE_COUNT = 0;
    ulStyle.transition = null;
    ulStyle.left = 100 + '%';
  }
  await fnDelay();
  ulStyle.transition = `left ${TIME_OPTION}ms linear`;
  ulStyle.left = ( -100 * SLIDE_COUNT ) + '%';
  await fnDelay(TIME_OPTION + 200);
  PERMISSION = true;
}

// 이전버튼 함수
const fnPrevAni = async () => {
  ulStyle.left = ( -100 * SLIDE_COUNT ) + '%';
  await fnDelay(TIME_OPTION);
  if(SLIDE_COUNT <= -1){
    SLIDE_COUNT = slideLen - 1;
    ulStyle.transition = null;
    ulStyle.left = ( -100 * SLIDE_COUNT ) + '%';
  }
  await fnDelay(200);
  ulStyle.transition = `left ${TIME_OPTION}ms linear`;
  PERMISSION = true;
};
```

```
// 이벤트 처리 -----

slideNext.addEventListener('click', (e) => {
  e.preventDefault();
  if(PERMISSION){
    PERMISSION = false;
    SLIDE_COUNT += 1;
    fnNextAni();
  };
});

slidePrev.addEventListener('click', (e) => {
  e.preventDefault();
  if(PERMISSION){
    PERMISSION = false;
    SLIDE_COUNT -= 1;
    fnPrevAni();
  };
});
```