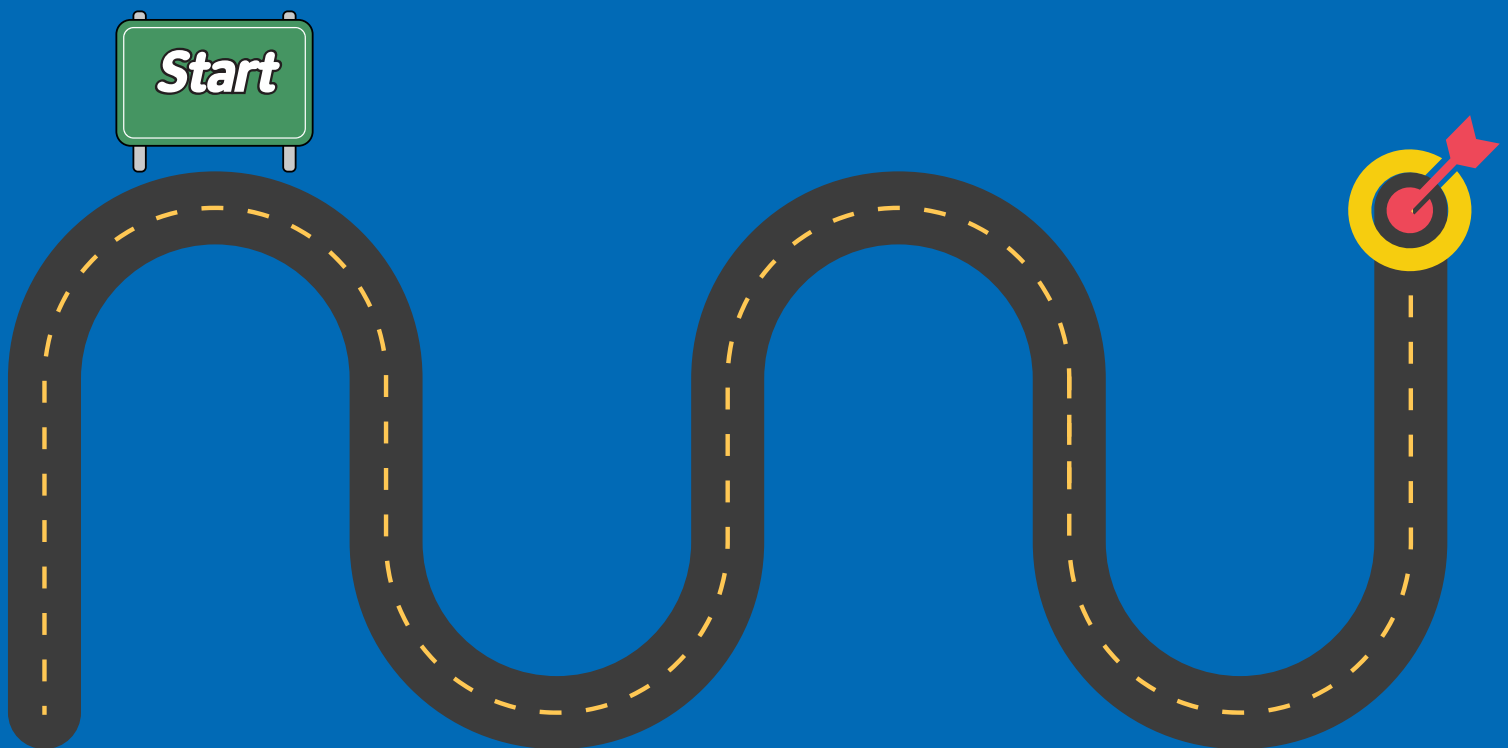


웹 프로그래밍 활용

HTML, CSS 기능구현



01

드롭다운메뉴

1-1 개념

드롭다운 메뉴의 개념

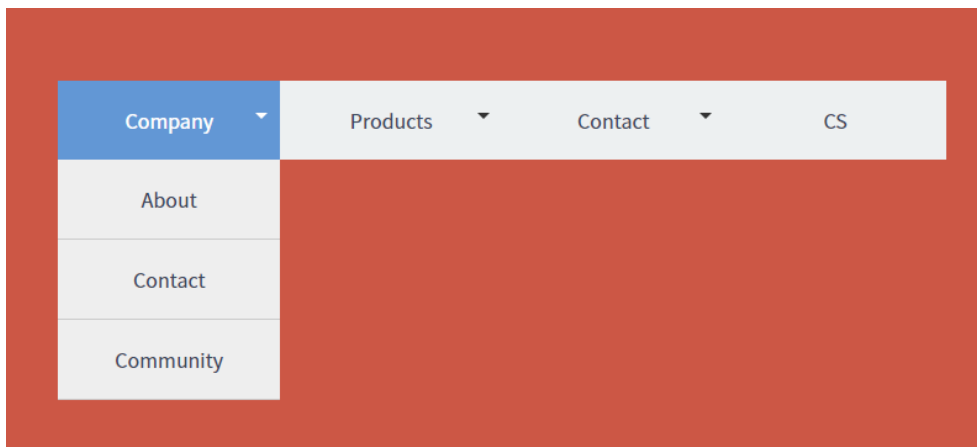
! 드롭다운메뉴

특정 메뉴의 숨겨진 하위 메뉴

01. 드롭다운메뉴이란?

사용자가 특정 메뉴 항목을 "클릭하거나, 마우스를 올렸을 때" 숨겨져있던 하위 메뉴 항목들이 나타남

** 하위 메뉴 항목들은 특정 메뉴와 관련이 있고 추가적인 메뉴임



02. 드롭다운 메뉴의 원리

- 1) 드롭다운메뉴는 특정 메뉴와 관련된 내용이니, 특정 메뉴 항목(li)과 같이 묶여있어야함
- 2) 드롭다운메뉴는 특정 메뉴 항목 바로 밑에 배치되어야하고 다른 메뉴 항목에 영향을 주지 않아야 함
- 3) 보이지 않게 숨겼다가 특정 조건(클릭하거나 마우스를 올렸을 때)을 만족하면, 나타나야함

03. 드롭다운 메뉴의 원리의 코드화

- 1) 드롭다운메뉴는 특정 메뉴와 관련된 내용이니, 특정 메뉴 항목(li)과 같이 묶여있어야함

```
<body>
  <ul class="main_menu">
    <li>Company
      <ul class="sub_menu">
        <li>About</li>
        <li>Contact</li>
        <li>Community</li>
      </ul>
    </li>
    <li>Products</li>
    <li>Contact</li>
    <li>CS</li>
  </ul>
</body>
```

특정 메뉴 항목

추가 메뉴 항목

2) 드롭다운메뉴는 특정 메뉴 항목 바로 밑에 배치되어야하고 다른 메뉴 항목에 영향을 주지 않아야 함

```
/* 특정 항목의 추가 메뉴 배치 작업*/
/* 특정 항목 li를 선택하여 하위 요소가
position:absolute; 했을 때의 기준점 제시 */
.main_menu > li:nth-child(1){
|   position: relative;
}

/* 특정 항목 li를 기준으로 position:absolute;
하고 위치 조정 : absolute해야 다른 메뉴 항목에
영향을 주지 않음*/
.sub_menu {
|   position: absolute;
|   top: 30px;
|   left: 0;
}
```

3) 보이지 않게 숨겼다가 특정 조건(클릭하거나 마우스를 올렸을 때)을 만족하면, 나타나야함

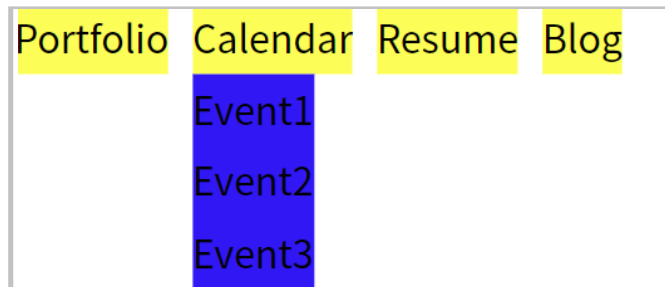
```
.sub_menu {
|   position: absolute;
|   top: 30px;
|   left: 0;
|   display: none;
|   /*배치 끝나면 시각,구조적 숨기기*/
}

/* 특정 조건 지정하고, sub_menu 나타나도록 속성
지정*/
.main_menu > li:nth-child(1):hover .sub_menu {
|   display: block; /*첫번째 항목에 마우스를
|   올렸을 때 숨겨져 있는 메뉴 나타나도록 지정
|   */
}
```

실습하기

[예제 1-1] 기본 드롭다운메뉴 구현 (dropdown.html)

```
<body>
  <ul class="main_menu">
    <li>Portfolio</li>
    <li>Calendar
      <ul class="sub_menu">
        <li>Event1</li>
        <li>Event2</li>
        <li>Event3</li>
      </ul>
    </li>
    <li>Resume</li>
    <li>Blog</li>
  </ul>
</body>
```



[기본 드롭다운메뉴 결과물]

[예제 1-1] 기본 드롭다운메뉴 구현 (dropdown.css)

```
*{
  padding: 0;
  margin: 0;
  box-sizing: border-box;
  text-decoration: none;
  list-style: none;
}

.main_menu{
  display: flex;
}

.main_menu > li {
  margin-right: 10px;
  height: 30px;
  line-height: 30px;
  background-color: yellow;
}

.main_menu > li:nth-child(2){
  position: relative;
}

.sub_menu {
  position: absolute;
  top: 30px;
  left: 0;
  display: none;
  background-color: blue;
}

.main_menu > li:nth-child(2):hover .sub_menu {
  display: block;
}
```

2-1 개념

라벨태그의 개념

💡 `<label>[/label>`

폼 요소에 대한 힌트를 개선하기 위해 사용되는 요소

01. label 태그란?

입력 양식 요소와 주로 사용되며, 입력 양식 요소의 목적을 명확하게 설명하거나 디자인, 스타일링 할 때 사용함.

02. label 태그 사용방법

- 특정 입력 필드와 연결 : for 속성 = "연결할 입력 필드의 id값"
- 연결한 결과 : 라벨을 클릭하면 해당 입력 필드에 포커스가 이동함

** 항상 잘 연결되었는지 확인 필요



실습하기

[예제 2-1] 라벨 연결하기 (label.html)

```
<body>
  <input type="radio"
name="gender" id="woman">
  <label for="woman">여자
</label>
  <input type="radio"
name="gender" id="man">
  <label for="man">남자</label>
</body>
```

☐ 여자 ☒ 남자

[기본 라벨 결과물]

2-2 개념

position:absolute:



position

요소의 배치 방법 설정

01. HTML 기본 배치 원칙

- 요소와 요소는 겹칠 수가 없다 ~> 유일한 예외 조건이 position:absolute;
- 블록 레벨 요소는 세로로 쌓이고, 인라인 레벨 요소는 가로로 쌓인다

02. position:absolute; 특징

- 1) 기준점 : 부모/조상 중에 position:relative; 값을 가진 요소를 기준, 없다면 body태그 기준
- 2) 기준점에 대한 top, bottom, left, right 속성과 값을 사용할 수 있음
- 3) 첫번째 기본 배치 원칙에 대한 예외 조건이므로, 요소와 요소가 겹칠 수 있어 z-index 속성을 쓸 수 있음
- 4) 요소끼리 겹칠 수 있다는 것은 문서 흐름에서 제거된 것임 (absolute된 요소의 빈 자리를 다른 요소가 채울 수 있음)
- 5) 가로,세로 중앙 정렬하기가 수월함



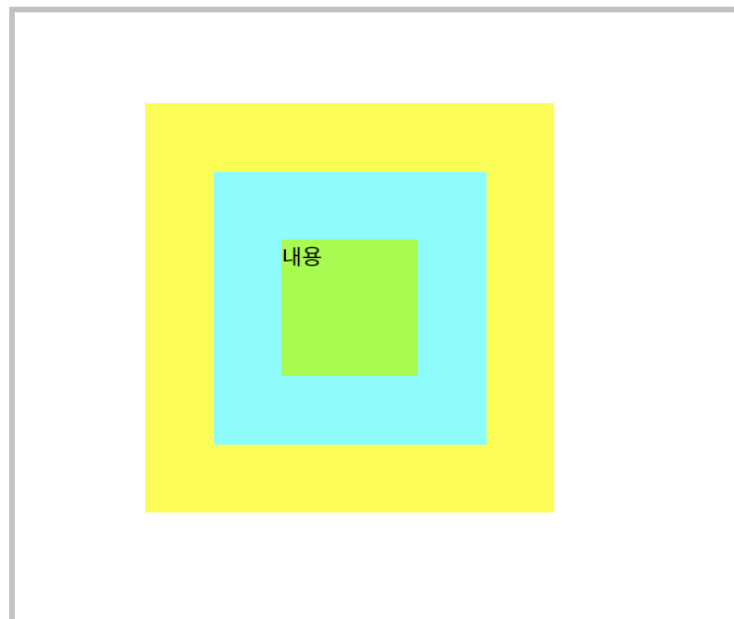
실습하기

[예제 2-2] position:absolute 사용하기 (absolute.html)

```
<body>
  <div class="level1">
    <div class="level2">
      <div class="level3">
        <p class="content">내용</p>
      </div>
    </div>
  </div>
</body>
```



```
*{
    margin: 0;
    padding: 0;
    text-decoration: none;
    list-style: none;
}
.level1{
    width: 300px;
    height: 300px;
    margin-top: 100px;
    margin-left: 100px;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: yellow;
}
.level2{
    width: 200px;
    height: 200px;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: aqua;
}
.level3{
    position: relative;
    width: 100px;
    height: 100px;
    background-color: chartreuse;
}
.content{
    position: absolute;
    top: 0%;
    left: 0%;
}
```



[position:absolute; 결과물]

2-3 개념

가로 정렬



display

정렬하려는 자식 요소들의 직계 부모 요소 선택, display:flex; 작성

** 특징

자식 요소들이 부모 요소보다 클 경우, 1) 자식 요소들이 짜부러지거나 2) flex-wrap 속성값 변경하여 줄 바꿈이 됨

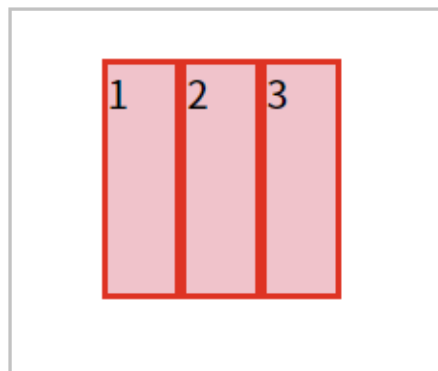
해당 특징은 슬라이더에서 불편함, 추가 작업 필요



실습하기

[예제 2-3] display:flex; 사용하기 (flex.html)

```
<body>
  <div class="slidebox">
    <div class="slides">
      <div class="slide">1</div>
      <div class="slide">2</div>
      <div class="slide">3</div>
    </div>
  </div>
</body>
```



[display:flex 결과물]

[예제 2-3] display:flex; 사용하기 (flex.css)

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.slidebox{
  width: 100px;
  height: 100px;
  margin: 30px auto;
  background-color: pink;
}

.slides{
  display: flex;
}

.slide{
  width: 100px;
  height: 100px;
  border: 3px solid red;
}
```




display

인라인요소의 특징을 가지면 됨, 정렬하려는 요소들 선택,
`display:inline-block` 작성

** 특징

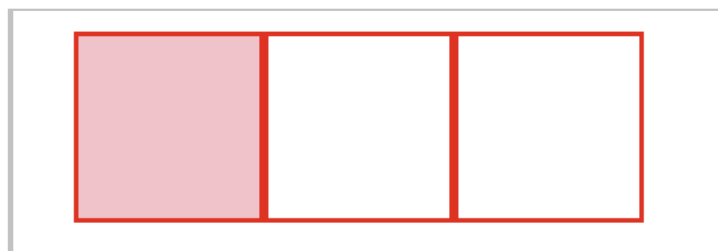
- 인라인레벨요소의 특징 (가로로 쌓임), 블록레벨요소의 특징 (크기 지정 가능) 두 특성을 가질 수 있음
- 단점은 자식요소가 부모요소보다 클 때 자동으로 줄바뀌짐 ~> `white-space: no-wrap`;을 줘야함
- `white-space:no-wrap`;으로 줄바꿈 하지 않는다고 바뀌도 요소와 요소 사이 지정하지 않은 공백이 발생함 (html 코드에서 줄바꿈이 공백으로 인식됨) ~> 해당 요소들의 부모에다가 `font-size:0`; 작성 필요



실습하기

[예제 2-4] `display:inline-block`; 사용하기 (flex2.css)

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
.slidebox{
    width: 100px;
    height: 100px;
    margin: 30px auto;
    background-color: pink;
    white-space: nowrap;
}
.slides {
    font-size: 0;
}
.slide{
    width: 100px;
    height: 100px;
    display: inline-block;
    border: 3px solid red;
}
```



[`display:inline-block` 결과물]

2-4 개념

일반 형제 선택자 (sibling)

형제 선택자

특정 요소의 형제 중에서 그 뒤에 위치한 모든 형제 요소를 선택

1. 사용법 : ~ 기호를 사용

ex) div ~ p : div태그의 뒤에 위치한 모든 형제 요소들 중 p태그

ex) input ~ .slidebox : input태그의 뒤에 위치한 모든 형제 요소들 중 class="slidebox"인 요소

실습하기

[예제 2-5] 형제 선택자 사용하기 (sibling.html)

```
<body>
  <div>
    <input type="checkbox" id="btn1">
    <label for="btn1">버튼 1</label>
    <div class="description">여기는 버튼 1의 설명입니다.</div>
  </div>
  <div>
    <input type="checkbox" id="btn2">
    <label for="btn2">버튼 2</label>
    <div class="description">여기는 버튼 2의 설명입니다.</div>
  </div>
  <div>
    <input type="checkbox" id="btn3">
    <label for="btn3">버튼 3</label>
    <div class="description">여기는 버튼 3의 설명입니다.</div>
  </div>
</body>
```

버튼 1

여기는 버튼 1의 설명입니다.

버튼 2

버튼 3

여기는 버튼 3의 설명입니다.

[형제 선택자 활용 결과물]



```
label {  
  display: inline-block;  
  padding: 5px;  
  margin: 5px 0;  
  background: #28a745;  
  color: white;  
  border-radius: 5px;  
  cursor: pointer;  
}  
  
.description {  
  margin-top: 5px;  
  padding: 5px;  
  border: 1px solid #ddd;  
  display: none;  
}  
  
input[type="checkbox"]:checked ~ .description {  
  display: block;  
}  
  
input[type="checkbox"] {  
  display: none;  
}
```

슬라이더의 개념



슬라이더

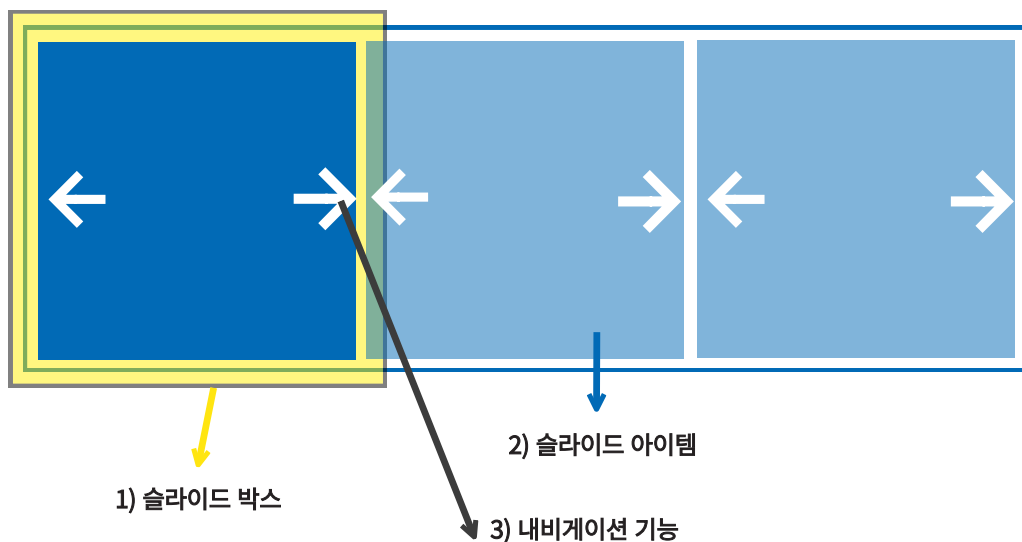
연속적인 콘텐츠를 하나의 콘텐츠로 보여주는 UI

01. 슬라이더란?

여러 이미지를 가로 또는 세로로 나열하여, 사용자가 한 번에 하나의 이미지나 콘텐츠를 볼 수 있게 함.
주로, 제품(상품) 및 이벤트 홍보나 주요 뉴스 정보 전달할 때 사용됨

02. 슬라이더의 구성

- 1) 슬라이드 박스: 콘텐츠가 실제로 보여지는 화면
- 2) 슬라이드 아이템: 각 슬라이드에 해당하는 콘텐츠(이미지, 텍스트 등).
- 3) 내비게이션 요소: 이전 및 다음 버튼(라벨) 등

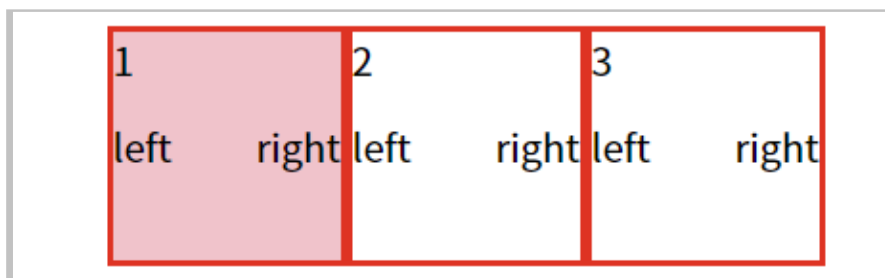


03. 슬라이더의 원리

- 1) 크기 지정
 - 슬라이드 박스 : 콘텐츠가 실제로 보여지는 화면의 크기 조정 (width)
 - 슬라이드 아이템 : 슬라이드 박스 사이즈의 100%로 크기 조정 (width)
- 2) 정렬 및 배치
 - 슬라이드 아이템들을 여백 없이 가로 정렬
 - 각 이미지별 라벨(왼쪽, 오른쪽) 위치 조정
- 3) 기능 구현
 - 콘텐츠별 input:radio가 있고 좌,우 라벨이 존재함. 좌,우 라벨이 콘텐츠의 input:radio와 연결됨을 확인 (리모컨으로 좌, 우 버튼을 눌러 TV 채널을 바꾸는 것과 같음)
 - n번째 콘텐츠의 radio버튼이 체크되었을시, 모든 콘텐츠를 동시 선택하여 한번에 (n-1)*100%만큼 왼쪽으로 이동



```
<body>
  <input type="radio" name="slide" id="s01">
  <input type="radio" name="slide" id="s02">
  <input type="radio" name="slide" id="s03">
  <div class="slidebox">
    <div class="slides">
      <div class="slide">
        <label for="s03" class="left">left</label>
        1
        <label for="s02" class="right">right</label>
      </div>
      <div class="slide">
        <label for="s01" class="left">left</label>
        2
        <label for="s03" class="right">right</label>
      </div>
      <div class="slide">
        <label for="s02" class="left">left</label>
        3
        <label for="s01" class="right">right</label>
      </div>
    </div>
  </div>
</body>
```



[slider 결과물]

[예제 2-6] 슬라이더 css 크기 지정 및 정렬 (slider.css)

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.slidebox{
    width: 100px;
    height: 100px;
    margin: 30px auto;
    background-color: pink;
    white-space: nowrap;
}

.slides {
    font-size: 0;
}

.slide{
    font-size: 16px;
    width: 100px;
    height: 100px;
    display: inline-block;
    border: 3px solid red;
}

```

[예제 2-6] 슬라이더 css label 배치하기 (slider.css)

```

.slide{
    font-size: 16px;
    width: 100px;
    height: 100px;
    display: inline-block;
    border: 3px solid red;
    position: relative;
}

label{
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
}

.left{
    left: 0;
}

.right{
    right: 0;
}

```

[예제 2-6] 슬라이더 css 기능부 (slider.css)

```

#s01:checked ~ .slidebox .slide{
    transform: translateX(0%);
}

#s02:checked ~ .slidebox .slide{
    transform: translateX(-100%);
}

#s03:checked ~ .slidebox .slide{
    transform: translateX(-200%);
}

input{
    display: none;
}

```

2-6 개념

고정 라벨 슬라이더

고정 라벨 슬라이더

콘텐츠를 전환해도 라벨은 고정된 자리에 위치한 슬라이더

01. 기존 슬라이더

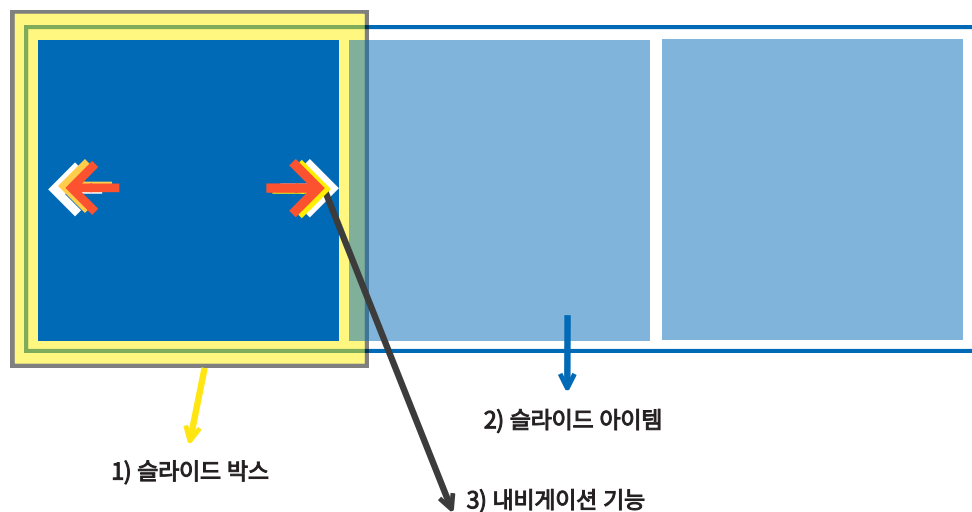
콘텐츠를 감싼 상자(div/li)를 기준으로 왼쪽/오른쪽 라벨을 각 콘텐츠별로 배치함

02. 고정 라벨 슬라이더

콘텐츠를 감싼 상자(div/li)의 기준이 아닌 콘텐츠를 보여주는 화면 즉, 슬라이드 박스 기준으로 왼쪽/오른쪽 라벨을 배치함

그렇게 되면, 콘텐츠별 라벨이 같은 위치에 겹쳐짐

그렇게 되면, 라벨에 의한 기능이 제대로 동작하지 않음 (라벨들이 겹쳐져있어서 정확히 어떤 라벨이 눌러졌는지 확인이 어려움)



03. 고정 라벨 슬라이더의 구성 (html)

- 1) 보여질 화면 : slidebox
- 2) 콘텐츠들의 묶음 : slides
- 3) 실제콘텐츠 : slide
- 4) 콘텐츠의 라벨들의 묶음 : controlbox
- 5) 콘텐츠별 라벨 묶음 : control01, control02, control03

04. 고정 라벨 슬라이더의 원리

1) 크기 지정

- 슬라이드 박스 : 콘텐츠가 실제로 보여지는 화면의 크기 조정 (width)
- 슬라이드 아이템 : 슬라이드 박스 사이즈의 100%로 크기 조정 (width)

2) 정렬 및 배치

- 슬라이드 아이템들을 여백 없이 가로 정렬
- "슬라이드 박스 기준으로" 라벨(왼쪽, 오른쪽) 위치 조정 : 차이점

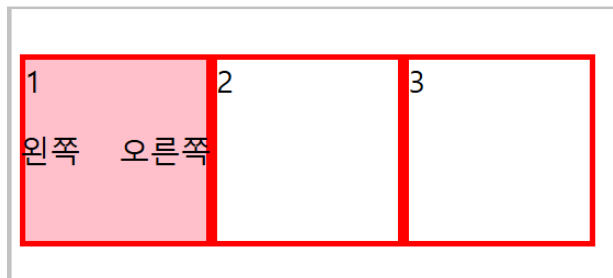
3) 기능 구현

- 콘텐츠별 input:radio가 있고 겹쳐져 있지만 콘텐츠별 좌,우 라벨이 존재함. 라벨 모두 display:none;
- 기본 checked를 설정해두고, checked에 해당하는 라벨만 display:block;
- n번째 콘텐츠의 radio버튼이 체크되었을시, 모든 콘텐츠를 동시 선택하여 한번에 (n-1)*100%만큼 왼쪽으로 이동

실습하기

[예제 2-7] 고정 라벨 슬라이더 html 기본 구성 (slider.html)

```
<body>
  <section id="slider">
    <input type="radio" name="slide" id="s01" checked>
    <input type="radio" name="slide" id="s02">
    <input type="radio" name="slide" id="s03">
    <div class="slidebox">
      <div class="slides">
        <div class="slide">1</div>
        <div class="slide">2</div>
        <div class="slide">3</div>
      </div>
      <div class="controlbox">
        <div class="control01">
          <label for="s03" class="left">왼쪽</label>
          <label for="s02" class="right">오른쪽</label>
        </div>
        <div class="control02">
          <label for="s01" class="left">왼쪽</label>
          <label for="s03" class="right">오른쪽</label>
        </div>
        <div class="control03">
          <label for="s02" class="left">왼쪽</label>
          <label for="s01" class="right">오른쪽</label>
        </div>
      </div>
    </div>
  </section>
```



[고정 라벨 slider 결과물]

[예제 2-7] 슬라이더 css 크기 지정 및 정렬 (slider.css)

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
#slider .slidebox {
    width: 100px;
    height: 100px;
    margin: 30px auto;
    background-color: pink;
    white-space: nowrap;
    position: relative;
}
#slider .slidebox .slides {
    font-size: 0;
}
#slider .slidebox .slides .slide {
    font-size: 16px;
    width: 100px;
    height: 100px;
    display: inline-block;
    transition: all .5s;
    border: 3px solid red;
}
```

[예제 2-7] 슬라이더 css label 배치하기 (slider.css)

```
/* 좌우로 넘기는 label에 대한 스타일 */
#slider .controlbox label {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    cursor: pointer;
    display: none;
}
#slider .controlbox .left {
    left: 0px;
}
#slider .controlbox .right {
    right: 0px;
}
```

[예제 2-7] 슬라이더 css label 기능부 (slider.css)

```
#slider input[id="s01"]:checked ~
.slidebox .control01 label{
    display: block;
}
#slider input[id="s02"]:checked ~
.slidebox .control02 label{
    display: block;
}
#slider input[id="s03"]:checked ~
.slidebox .control03 label{
    display: block;
}
```



#slider	#slider
input[id="s01"]:checked~.slidebox .slides	input[id="s03"]:checked~.slidebox .slides
.slide {	.slide {
transform: translateX(0%);	transform: translateX(-200%);
}	}
#slider	/* input 가리기 */
input[id="s02"]:checked~.slidebox .slides	#slider input[type="radio"] {
.slide {	display: none;
transform: translateX(-100%);	}
}	

3-1 개념

탭의 개념



탭

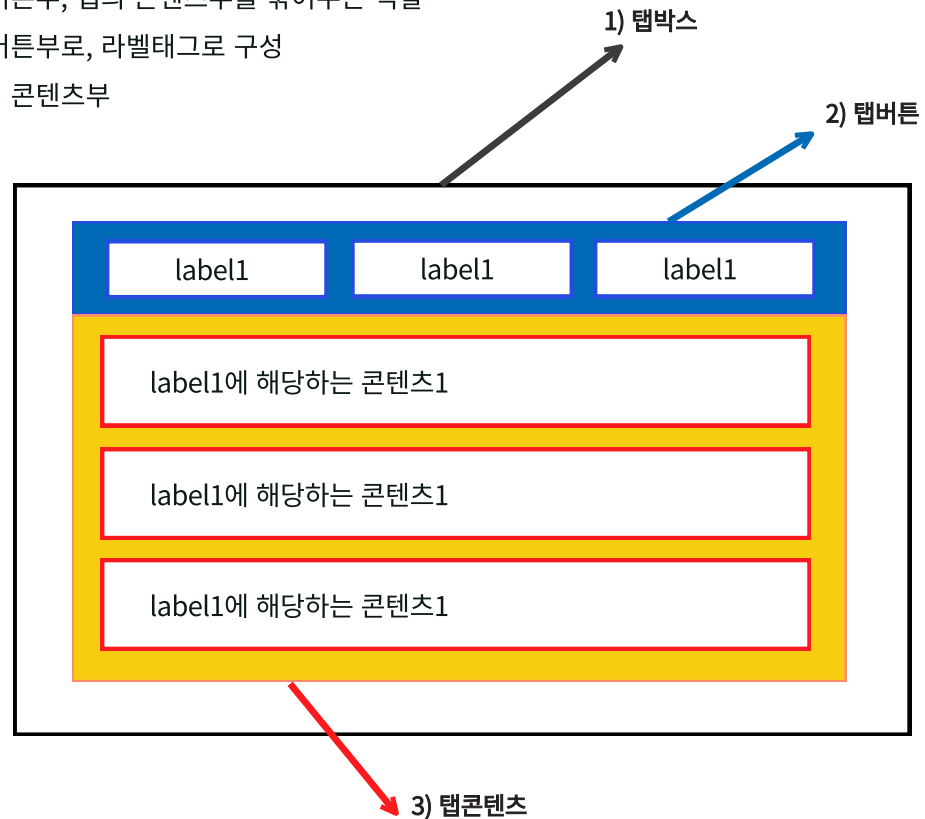
여러 개의 콘텐츠를 카테고리에 맞게 쉽게 탐색할 수 있도록 해주는 요소

01. 탭이란?

많은 양의 정보를 "카테고리별로" 나누어 보여줄 때, 여러 항목을 비교해야 할 때 주로 사용함.

02. 탭의 구성

- 탭박스 : 탭의 버튼부, 탭의 콘텐츠부를 묶어주는 역할
- 탭버튼 : 탭의 버튼부로, 라벨태그로 구성
- 탭콘텐츠 : 탭의 콘텐츠부



03. 탭의 원리 (html)

- 1) 카테고리별 input:radio 만들기
- 2) 탭버튼, 탭콘텐츠를 포함하는 탭박스 만들기
- 3) 탭버튼 안에 input:radio 각각에 연결되는 label 만들기
- 4) 탭콘텐츠 안에 모든 콘텐츠들 작성하기

04. 탭의 원리 (css)

1) 크기 지정

- 탭 박스 : 탭 라벨과 콘텐츠가 보여지는 상자의 크기 조정 (width)
- 라벨 크기 지정 : 균등하게 너비를 가지려면 (100/라벨갯수)%를 주면 됨 (width)

2) 기능 구현

- 콘텐츠 모두 없애기 display:none;
- 특정 콘텐츠 나타내기 : 특정 카테고리 input:radio가 checked 되었다면 특정 카테고리에 맞는 콘텐츠들 모두 display:block; 으로 나타내기
- 라벨의 배경색상 바꾸기 : 어떤 카테고리가 선택되었는지 티가 안나므로 배경색상 변경할 것. 특정 카테고리 input:radio가 checked 되었다면, 해당하는 라벨을 선택하여 background-color 변경

실습하기

[예제 3-1] 기본 탭 html 기본 구성 (tab.html)

```
<body>
  <input type="radio" name="tab_radio" id="a">
  <input type="radio" name="tab_radio" id="b">
  <input type="radio" name="tab_radio" id="c">
  <input type="radio" name="tab_radio" id="d">
  <div id="tab_box">
    <div id="tab_button">
      <label for="a" class="a">aaa</label>
      <label for="b" class="b">bbb</label>
      <label for="c" class="c">ccc</label>
      <label for="d" class="d">ddd</label>
    </div>
    <div id="tab_content">
      <div class="all a">첫번째 상자입니다.</div>
      <div class="all b">두번째 상자입니다.</div>
      <div class="all c">세번째 상자입니다.</div>
      <div class="all d">네번째 상자입니다.</div>
    </div>
  </div>
</body>
```



[기본 탭 결과물]

[예제 3-1] 기본 탭 css 크기 지정 (tab.css)

```
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
#tab_box{
    width: 60%;
    margin: 30px auto;
    border: 1px solid red;
}
#tab_button{
    font-size: 0px;
}
#tab_button label{
    width: 25%;
    display: inline-block;
    font-size: 16px;
    color: white;
    background-color: rgb(89, 89, 245);
    padding: 20px 0;
}
```

[예제 3-1] 기본 탭 css 콘텐츠 숨기고 나타내기 (tab.css)

```
.all {display: none;}
#a:checked ~ #tab_box #tab_content .a{
    display: block;
}
#b:checked ~ #tab_box #tab_content .b{
    display: block;
}
#c:checked ~ #tab_box #tab_content .c{
    display: block;
}
#d:checked ~ #tab_box #tab_content .d{
    display: block;
}
```

[예제 3-1] 기본 탭 css 특정 label 배경색 바꾸기 (tab.css)

```
#a:checked ~ #tab_box #tab_button .a{
    background-color: pink;
}
#b:checked ~ #tab_box #tab_button .b{
    background-color: pink;
}
#c:checked ~ #tab_box #tab_button .c{
    background-color: pink;
}
#d:checked ~ #tab_box #tab_button .d{
    background-color: pink;
}
#a,#b,#c,#d{
    display: none;
}
```