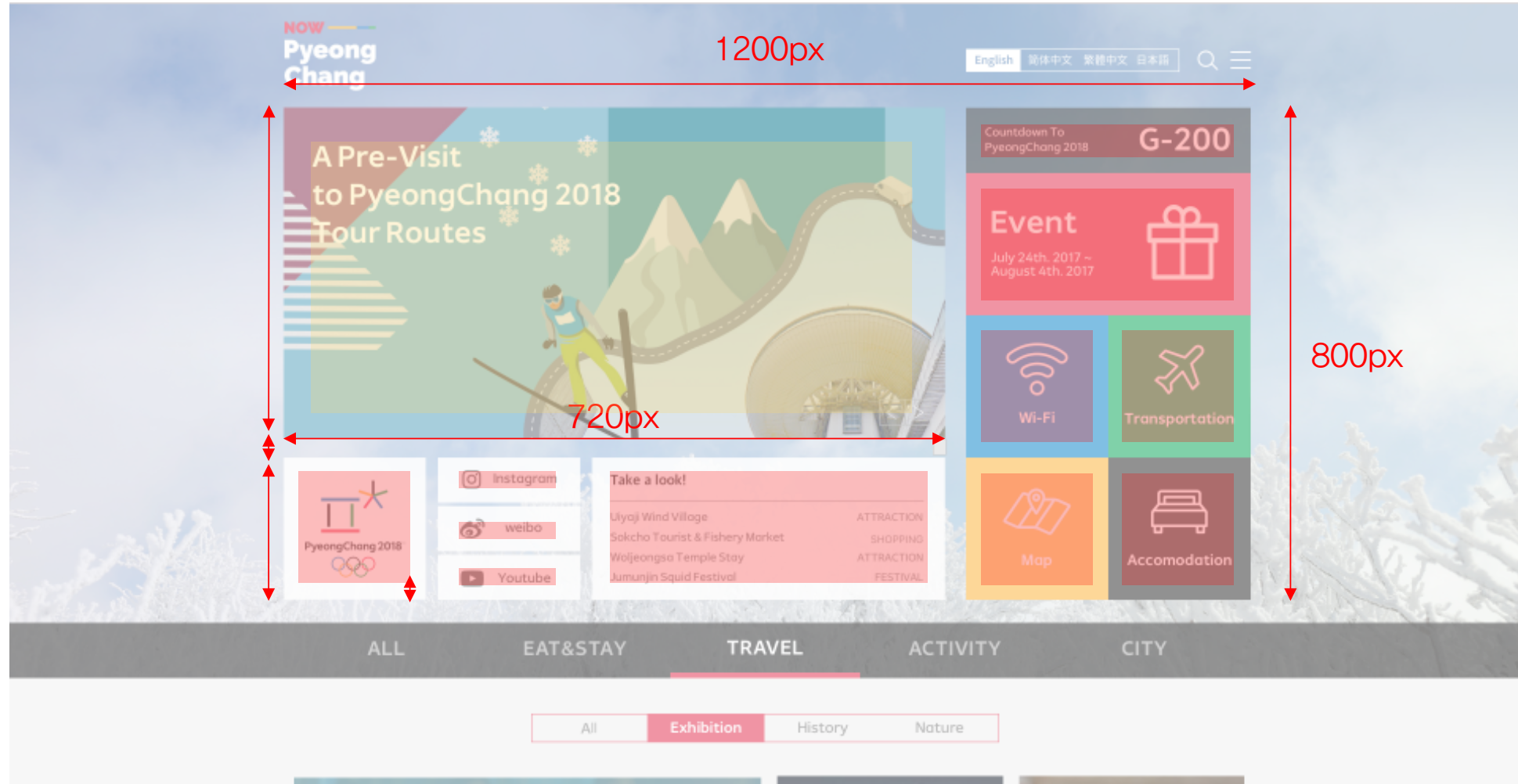


2주차 CSS심화 & 반응형 웹

2017.09.30

조성현

지난 주 실습



이번 주 실습

〈자기소개 페이지 만들기〉

자신이 마음에 드는 레퍼런스 사이트를 찾아,
적절하게 자신의 사이트로 만들기

필수 포함 사항

자신의 사진



이미지 넣는 방법

1. img태그
2. css의 background속성

▶ 이미지 넣는 방법

img태그 사용

```

```

이미지 파일 / url

설명

이미지 크기



이미지 넣는 방법

css background 속성 사용

```
<div class= "bg_image" > </div>
```

```
.bg_image {
```

```
    background: url( '/image/smile.gif' ) no-repeat;
```

```
    background-size: contain;
```

```
    width: 100px;
```

```
    height: auto;
```

```
}
```



이미지 넣는 방법

css background 속성 사용

```
<div class= “bg_image” > </div>
```

```
.bg_image {
```

```
    이미지 파일 / url      background: url( ‘/image/smile.gif’ ) no-repeat;
```

```
    background-size: contain;
```

```
    width: 100px;
```

```
    height: auto;
```

```
}
```



이미지 넣는 방법

css background 속성 사용

```
<div class= “bg_image” > </div>
```

```
.bg_image {
```

```
background: url( ‘/image/smile.gif’ ) no-repeat;
```

size속성 **background-size: contain;**

```
width: 100px;
```

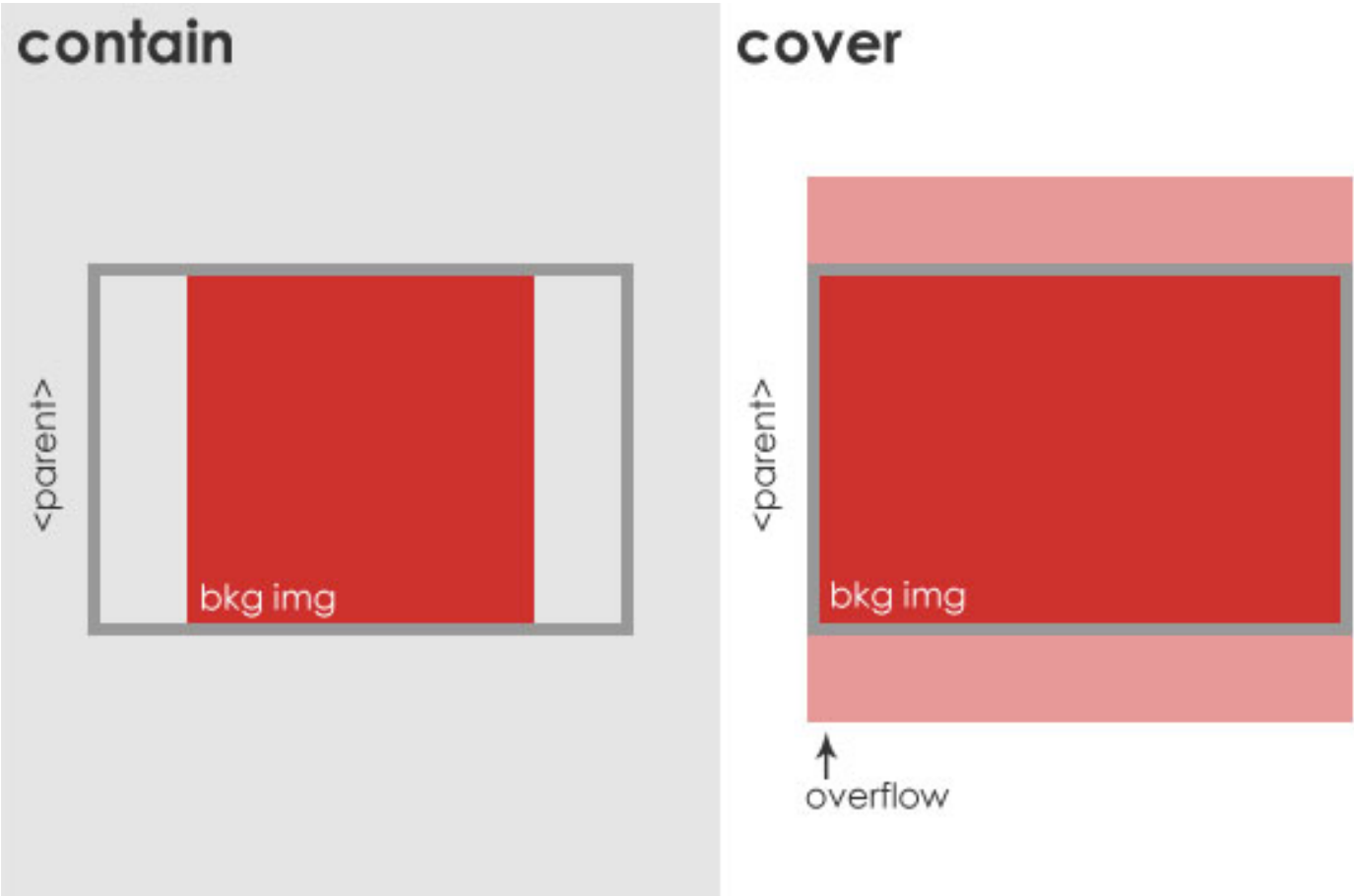
```
height: auto;
```

```
}
```




이미지 넣는 방법

css background 속성 사용



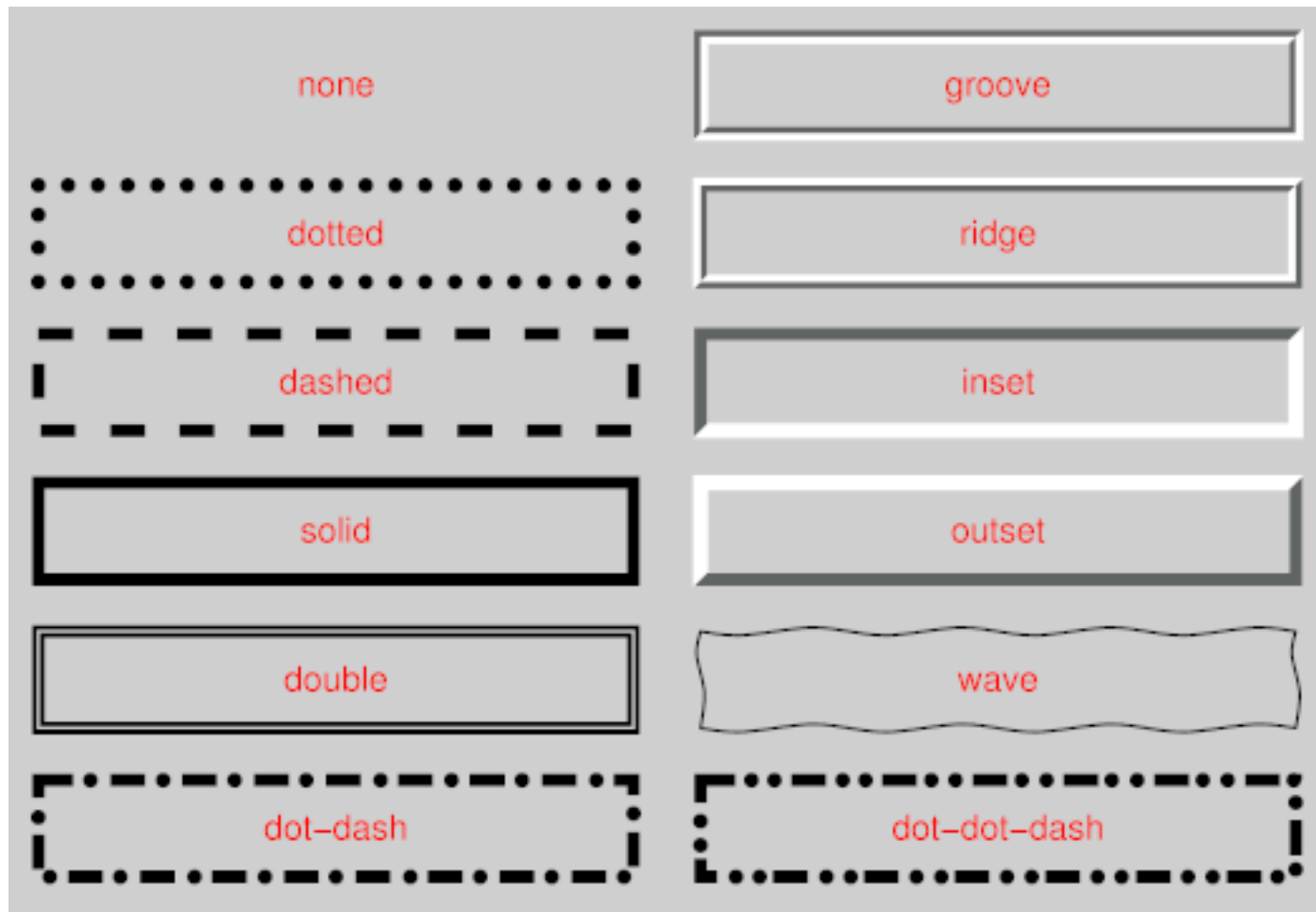
▶ 이미지 넣는 방법

img태그 vs css background속성

| img 태그 | css background 속성 |
|---|---|
| <ul style="list-style-type: none">• 사이즈 조정 시, 비율이 달라질 수 있음• 같은 태그 안에 text넣을 수 없음.• height: auto => 태그가 이미지 height를 알고 있음 | <ul style="list-style-type: none">• 사이즈를 조정해도 비율 유지 가능• background로 지정하기에 text를 넣을 수 있음. ex) <code><div class= "bg_img" > hello </div></code>• 이미지의 height를 모름. |

CSS속성

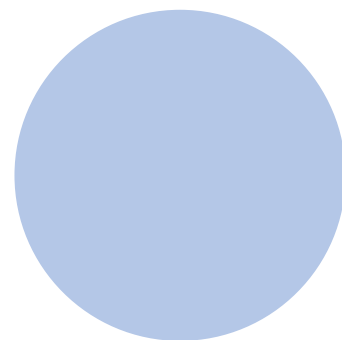
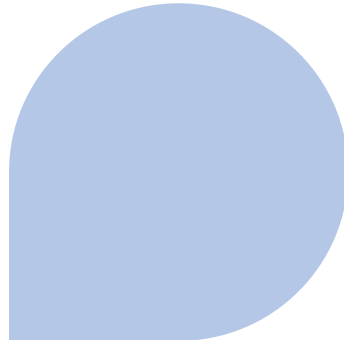
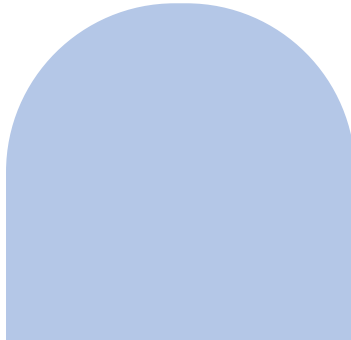
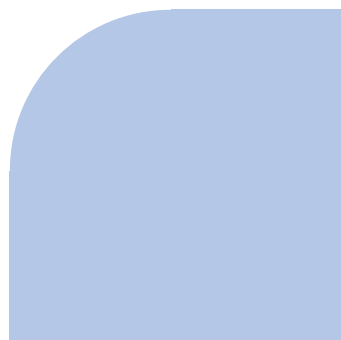
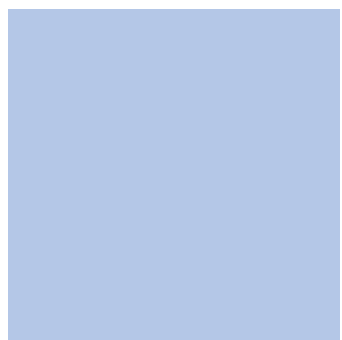
border-style





CSS속성

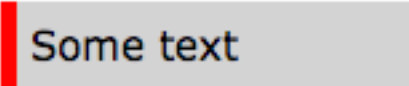
border-radius





CSS속성

border-top, right, bottom, left



```
p {  
  border-left: 6px solid red;  
  background-color: lightgrey;  
}
```



```
p {  
  border-bottom: 6px solid red;  
  background-color: lightgrey;  
}
```



CSS속성

border

border-style: solid;

border-width: 5px;

border-color: #000;



border: solid 5px #000;

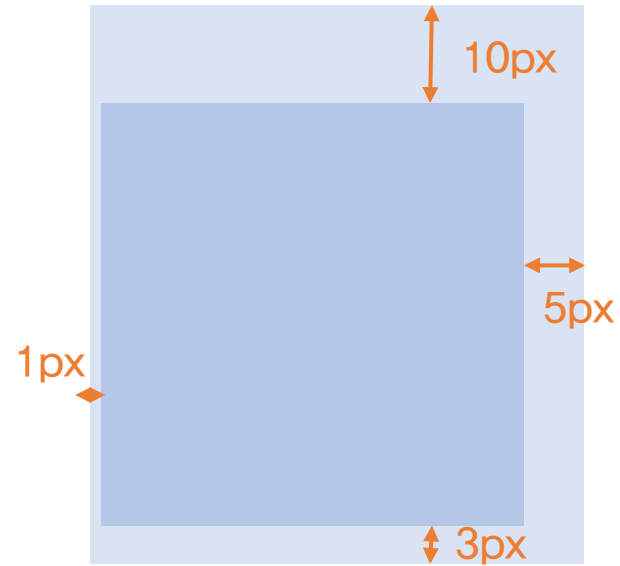
CSS속성

속성 적용 방향

상 우 하 좌

margin: 10px 5px 3px 1px;

padding도 동일한 방향으로!



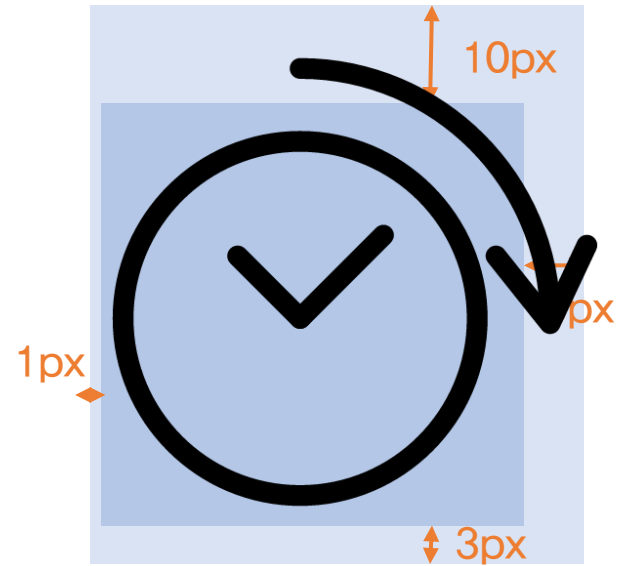
CSS속성

속성 적용 방향

상 우 하 좌

margin: 10px 5px 3px 1px;

padding도 동일한 방향으로!



페이지 꾸미기

Google Fonts

<https://fonts.google.com/>

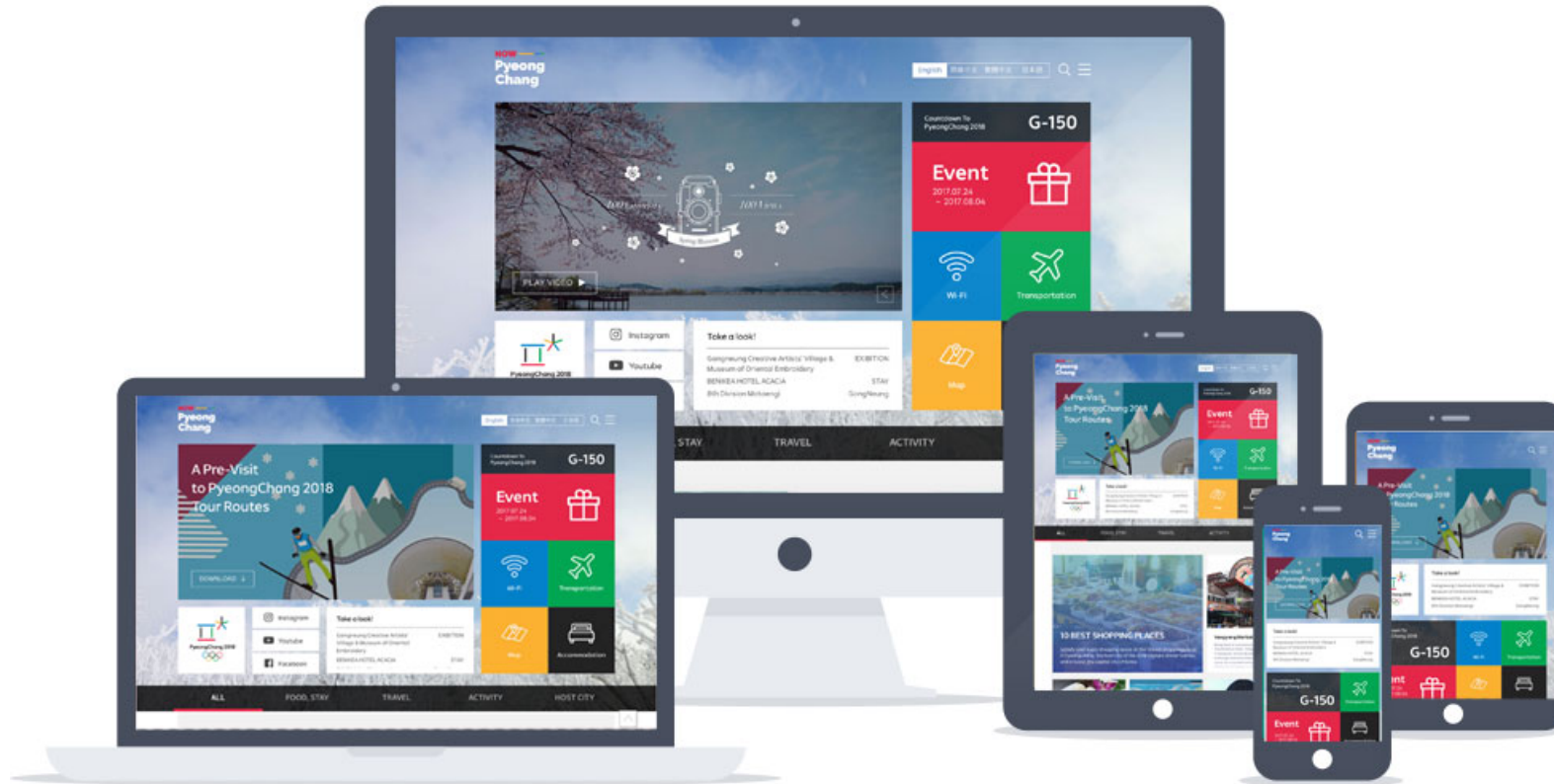


MATERIAL DESIGN

<https://material.io/icons/>

반응형 웹

웹 디자인 기법 중 하나. 웹(Web)에 접속하는 디바이스에 반응하는(Responsive) 디자인(Design)





반응형 웹

미디어쿼리

@media only all and (조건문) {실행문}



반응형 웹

미디어쿼리

@media only all and (조건문) {실행문}

미디어 쿼리가 시작됨을 선언.



반응형 웹

미디어쿼리

@media **only** all and (조건문) {실행문}

- 미디어 쿼리를 지원하는 사용자 에이전트만
미디어 쿼리 구문을 해석하라는 명령이며 생략 가능
- 생략 시 default 는 only



반응형 웹

미디어쿼리

@media only **all** and (조건문) {실행문}

- 미디어 쿼리를 해석해야 할 대상 미디어를 선언한 것
- all : 모든 미디어가 이 구문을 해석해야 한다.
- screen이나 print와 같은 특정 미디어를 언급할 수 있음.
- 생략 시 default는 all

반응형 웹

미디어쿼리

@media only all and (조건문) {실행문}

- 논리적으로 ‘AND’ 연산을 수행하여 앞과 뒤의 조건을 모두 만족해야 한다는 것을 의미
- 조건이 유일하거나 또는 only, all과 같은 선행 키워드가 생략되면 and 키워드는 사용하지 말아야 한다.
- and 대신 콤마 ‘,’ 기호를 사용하면 ‘OR’ 연산을 수행한다.
- ‘OR’ 연산은 나열된 조건 중에서 하나만 참이어도 {실행문}을 해석한다.



반응형 웹

미디어쿼리

@media only all and (조건문) {실행문}

- 조건문이 참일때{실행문}을 처리하고 거짓일 때 무시
- 조건문은 두 가지 이상 등장할 수 있다.
- 둘 이상의 조건문은 'and' 키워드 또는 콤마 ',' 기호로 연결해야 한다.



반응형 웹 미디어쿼리

@media only all and (조건문) {실행문}

- 일반적인 CSS 코드를 이 괄호 안에 작성한다.
- 브라우저는 (조건문)이 참일때 실행문 안쪽에 있는 CSS 코드를 해석한다.

반응형 웹

미디어쿼리

```
/* iPads (portrait and landscape) */
@media only screen
and (min-width : 768px)
and (max-width : 1024px) {
  body { background-color: orange; }
  h1 { color: black; }
}

/* iPads (landscape) */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : landscape) {
  body { background-color: black; }
  h1 { color: yellow; }
}

/* iPads (portrait) */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : portrait) {
  body { background-color: yellow; }
  h1 { color: black; }
}

/* Desktops and laptops */
@media only screen
and (min-width : 1025px) {
  body { background-color: blue; }
  h1 { color: yellow; }
}
```



CSS방법론 : BEM

Block **E**lement **M**odifier

CSS방법론 : BEM

Block Element Modifier

작명규칙(Naming Convention)

- 개발, 디버깅, 유지보수를 위하여 CSS 선택자의 이름을 가능한 한 명확하게 만드는 것이 목표이다.
- 소문자, 숫자 만을 이용해서 작명한다.
- 여러단어의 조합은 하이픈(-)으로 연결하여 작명한다.

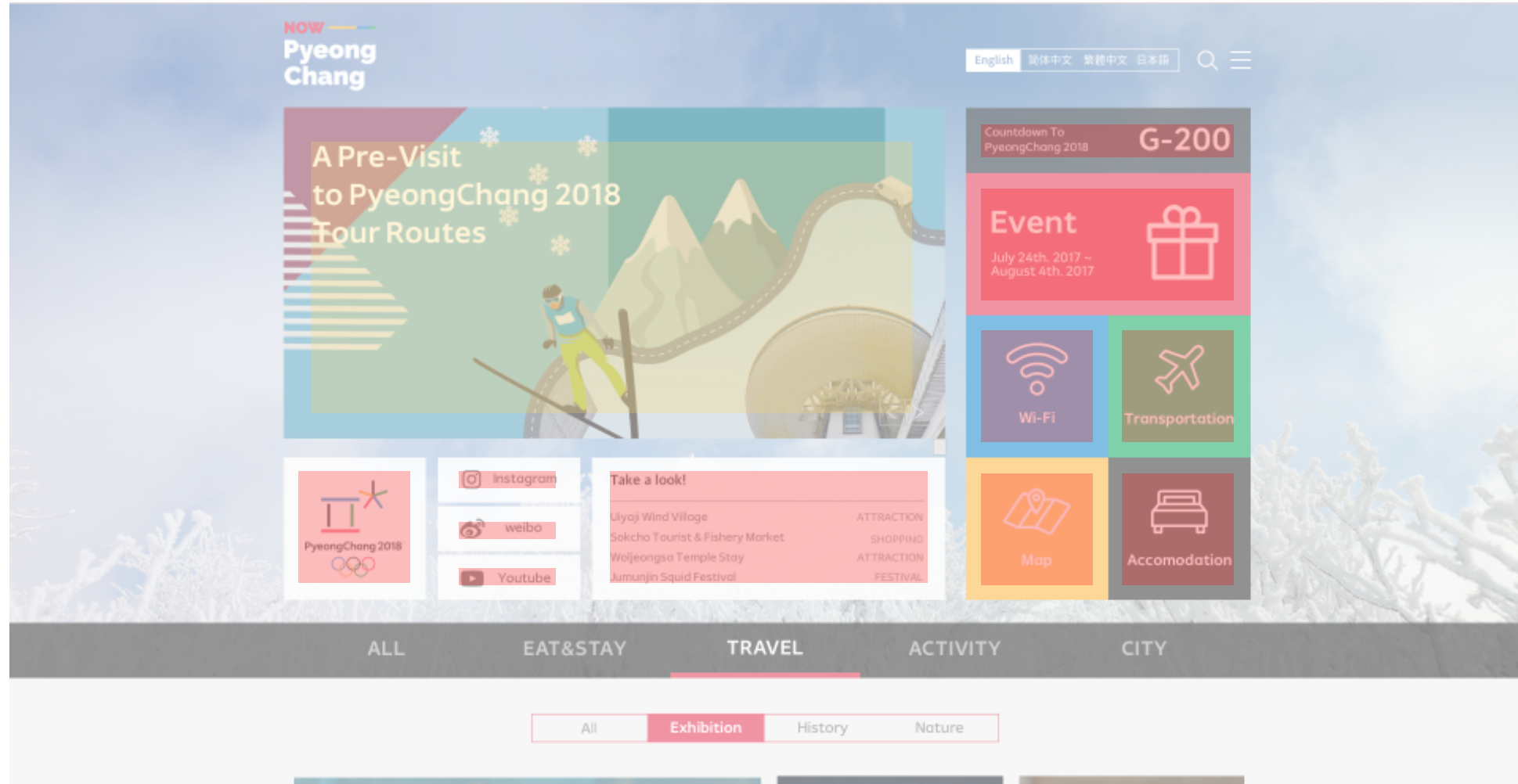
CSS방법론 : BEM

블록(Block)

- 재사용 할 수있는 기능적으로 독립적인 페이지 구성 요소. (HTML에서 블록은 class 속성으로 표시)
- 형태(red, big)가 아닌 목적(menu, button)에 맞게 결정해야 한다.
- 블록은 환경에 영향을 받지 않아야 한다. 즉, 여백이나 위치를 설정하면 안된다.
- 태그, id 선택자를 사용하면 안된다.
- 블록은 서로 중첩해서 작성 할 수 있다.

예) header, menu, search-form

CSS방법론 : BEM

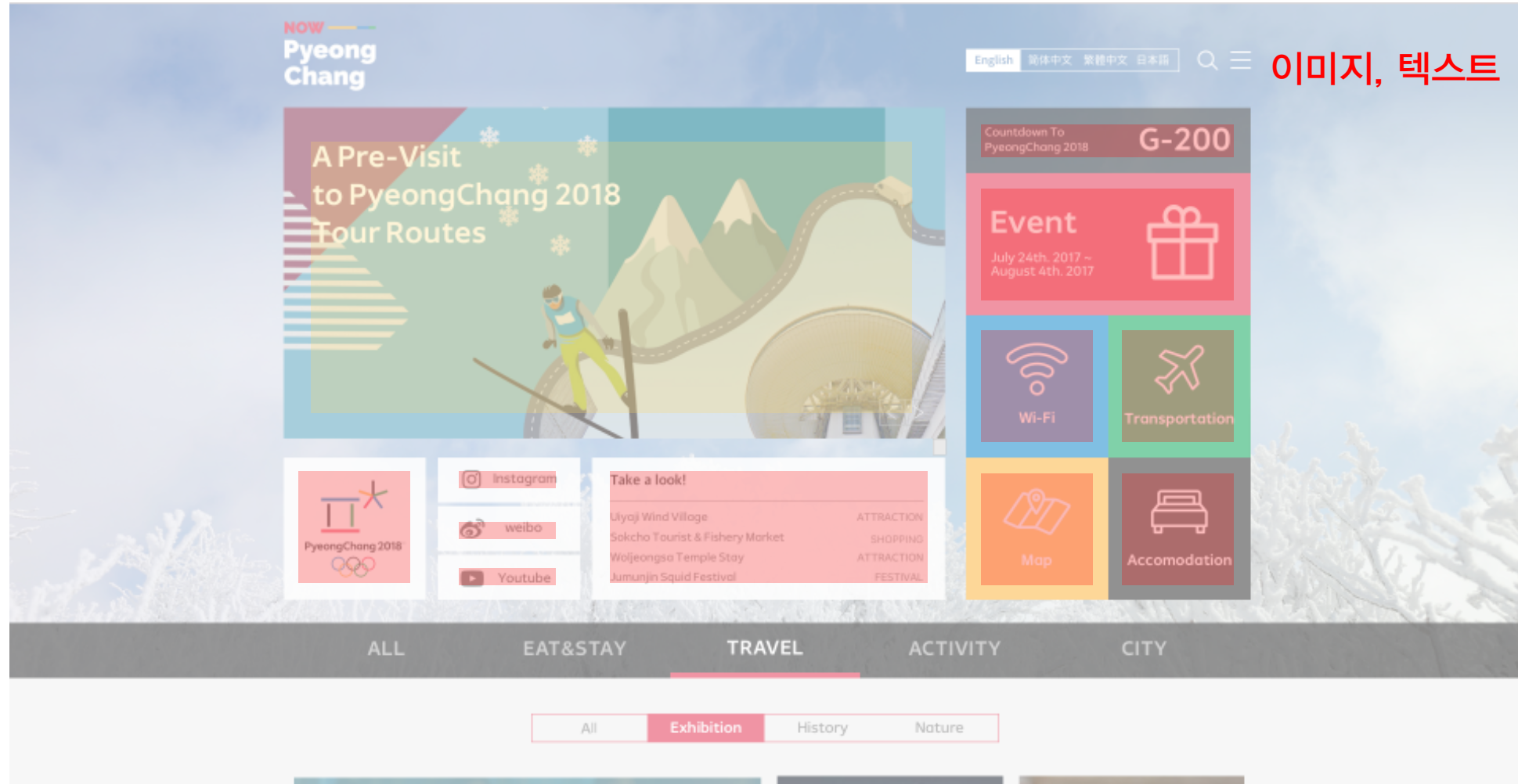


CSS방법론 : BEM

요소(Element)

- 블록안에서 특정 기능을 담당하는 부분.
- block__element 형태로 사용 (더블 언더바)
- 형태(red, big)가 아닌 목적(item, text, title)에 맞게 결정해야 한다.
- 요소는 중첩해서 작성 할 수 있다.
- 요소는 블록의 부분으로만 사용 할 수 있고 다른 요소의 부분으로 사용할 수 없다.
- 모든 블록에서 요소는 필수가 아닌 선택적으로 사용한다. 즉 블록안에 요소가 없을 수도 있다.
- 예) menu__item, header__title ...

CSS방법론 : BEM



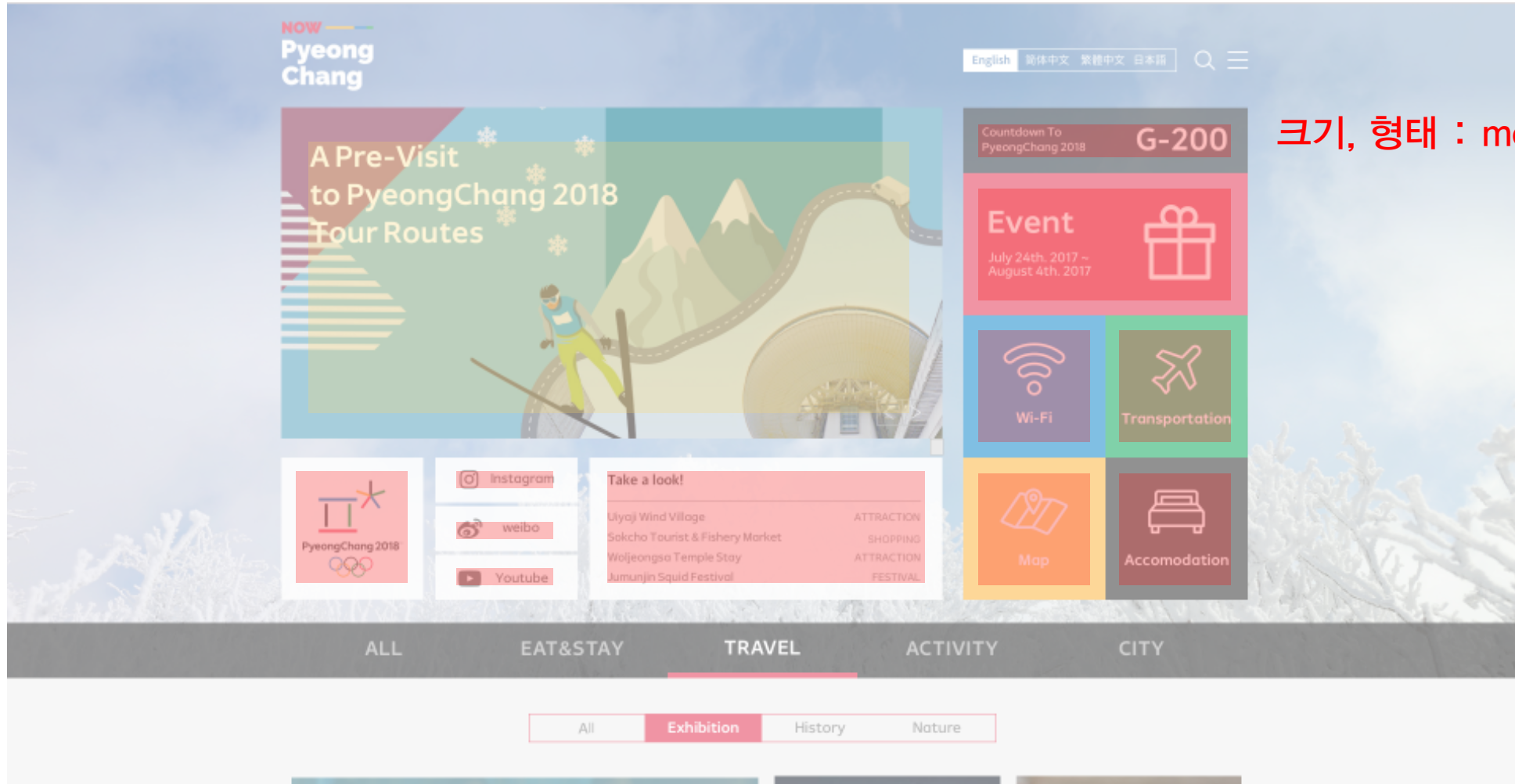
CSS방법론 : BEM

수식어(Modifier)

- 블록이나 요소의 모양(color, size..), 상태(disabled, checked..)를 정의한다.
- block__element — modifier, block — modifier 형태로 사용(더블 하이픈)
- 수식어는 단독으로 사용할 수 없다.

즉, 기본 블록과 요소에 추가하여 사용해야 한다. (class=" block__element block__element — modifier")

CSS방법론 : BEM



크기, 형태 : modifier

CSS방법론 : BEM

특징

- 각 클래스마다 기능이 분리되어 있기에, 중복되는 CSS를 줄일 수 있다.
- 클래스 이름만 보고 직관적으로 모양, 기능 등을 알 수 있음
- 네이밍이 수월해진다.
- 규칙을 정하기가 편하다.