



클론 코딩(2일차)

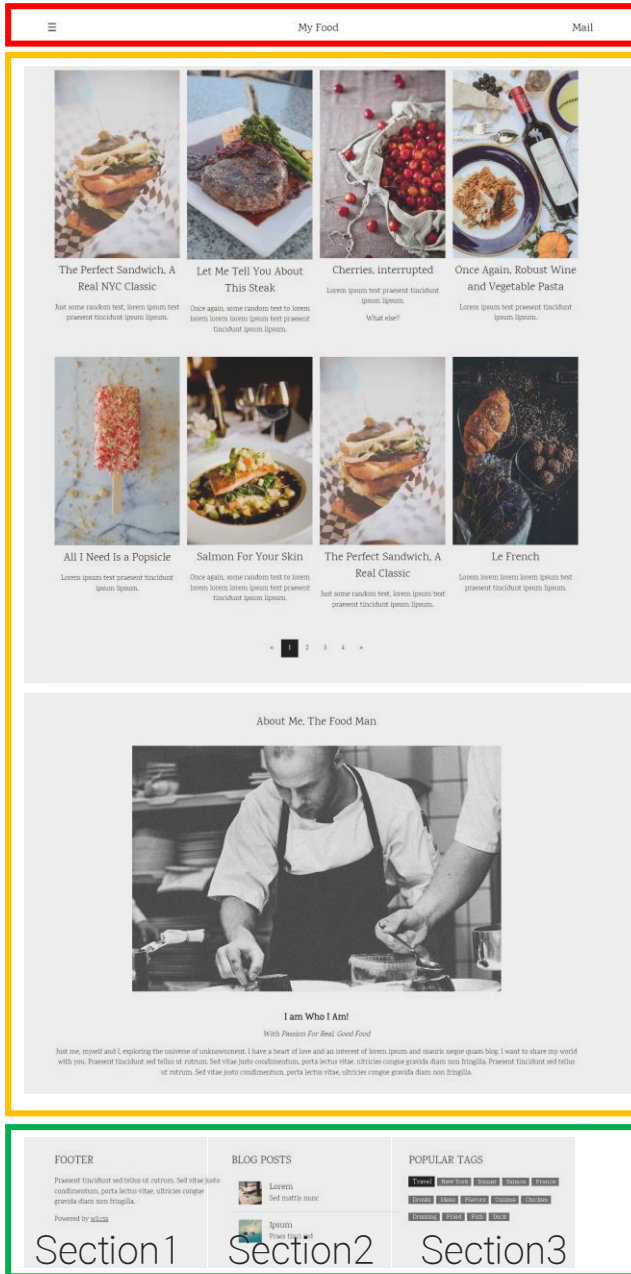
풀스택 자바 웹개발자 오후반
방미소



목차

1. 레이아웃 정의
2. 레이아웃 상세
3. 구현 결과
4. 소요시간 및 후기

1. 레이아웃 정의 (전체)



Section1

Section2

FOOTER

Present tuncidunt sed tellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla.

Powered by [joomla](#)

BLOG POSTS

 Lorem
Sed mattis enim.

 Ipsum
Pram ligula.

POPULAR TAGS

[Food](#) [Recipe](#) [Dessert](#) [Beverage](#) [Drink](#)

[Bread](#) [Salad](#) [Smoothie](#) [Juice](#) [Sauce](#)

Section 1

Section 2

Section 3

헤더

- 브라우저를 스크롤해도 상단에 고정

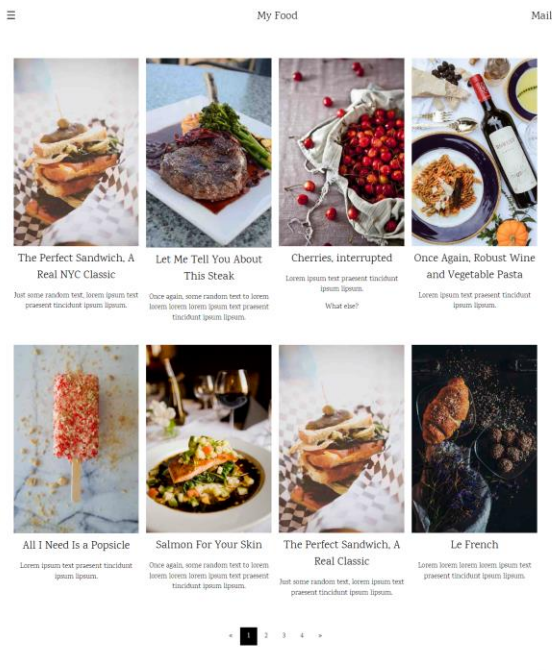
메인

- 크게 2개의 section으로 구분

푸터

- 크게 3개의 section으로 구분

1. 레이아웃 정의 (미디어쿼리)



미디어 쿼리

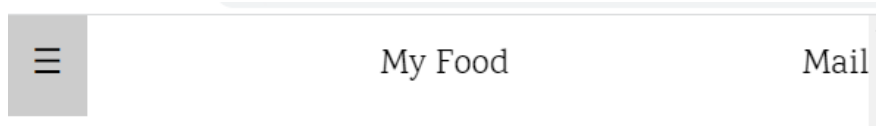
0~600px

- › 모든 요소 block 상태
- › 이미지가 브라우저 사이즈에 맞춰 확대

601px~

- › Section1 의 article들이 2행 4열로 보임

2. 레이아웃 상세 (헤더, 푸터)



헤더

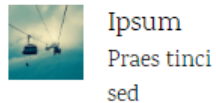
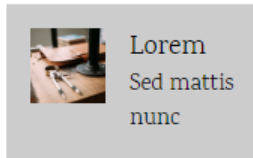
- 메뉴바는 브라우저를 스크롤해도 상단에 고정 (fixed)
- 왼쪽 메뉴 버튼
 - javascript 대신 페이지 상단 하이퍼링크 적용 (a href='#')
 - 마우스 위치 시 회색 박스 효과 (hover)
- 중간 타이틀, 우측 mail은 페이지 상단 하이퍼링크 적용 (a href='#')

FOOTER

Praesent tincidunt sed tellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla.

Powered by [w3.css](#)

BLOG POSTS



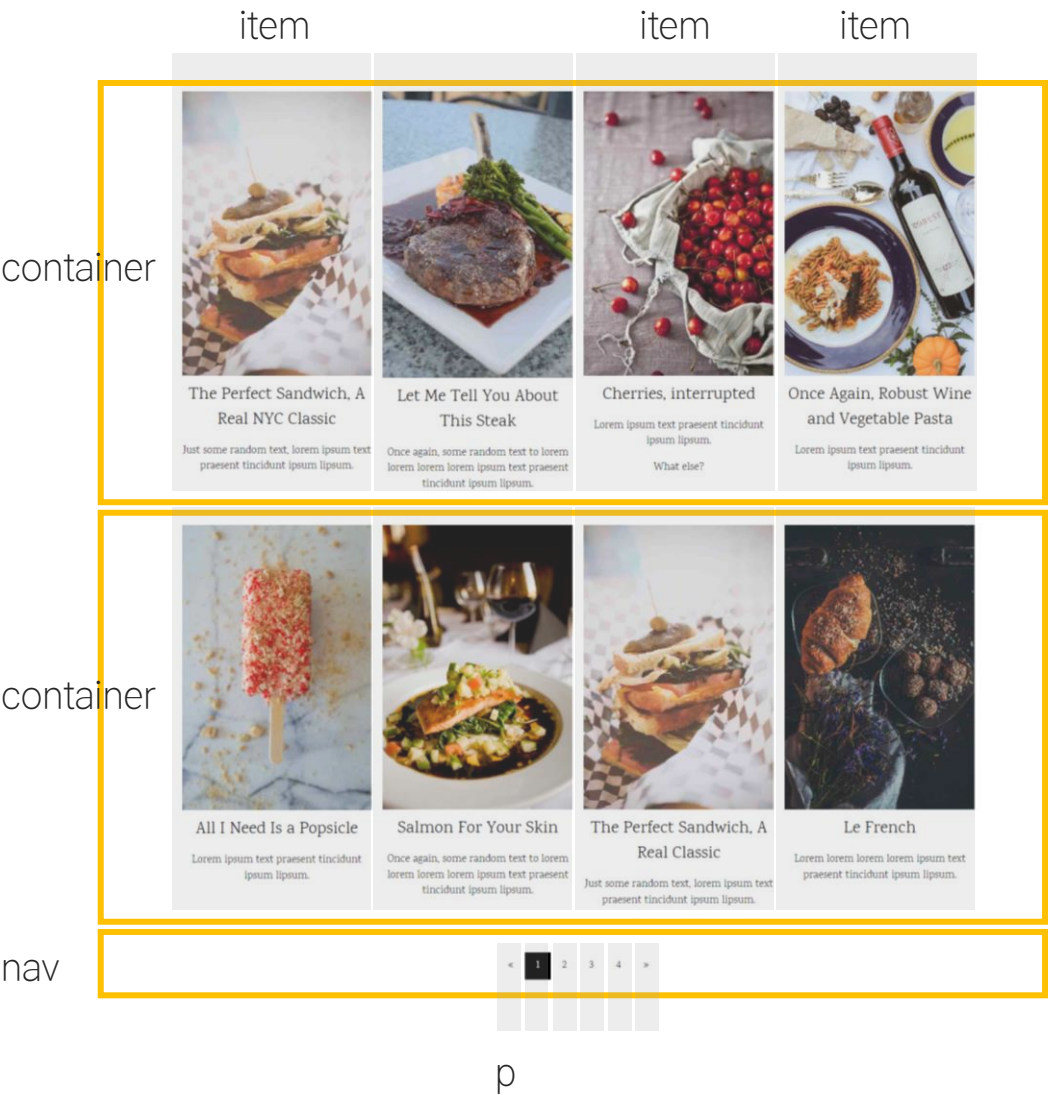
POPULAR TAGS



푸터

- 각각 width가 33.3%인 3개의 section으로 구성
- 중간 section은 내용물이 다시 별도의 배열로 정렬 + 회색 박스 hover 효과
- 우측 section 중 첫번째 tag는 검은 박스로 강조

2. 레이아웃 상세 (메인)



image



title

The Perfect Sandwich, A
Real NYC Classic

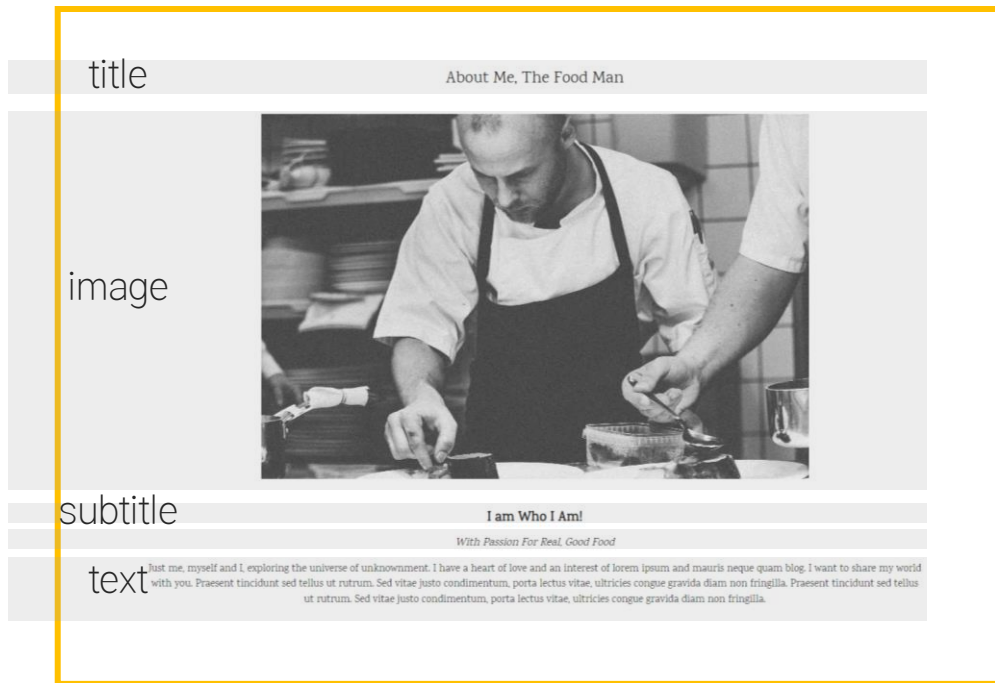
text

Just some random text, lorem ipsum text
praesent tincidunt ipsum ipsum.

메인 – section 1

- 지난 과제에서 미디어 쿼리 적용 시 item 정렬이 이상하게 뒤틀리던 현상은 container로 item을 한 번 더 감싸주어 해결할 수 있는 것을 발견
- 8개의 item을 각각 4개씩 2개의 container로 분류하여 배열
 - > 기본적으로는 아무 정렬 없이 모든 item block 표시 → 기본 상태
 - > 미디어 쿼리로 브라우저 사이즈 확장 시
 - + container 레벨에서 1차 정렬 (여기서는 2행 수직 정렬)
 - + container 안, item 레벨에서 2차 정렬 (4열 수평 정렬)
 - + 이 때 item은 width:25% 적용하여 균등 배열되도록 함
- nav 안의 각 아이템은 p로 작성한 후 수평 정렬

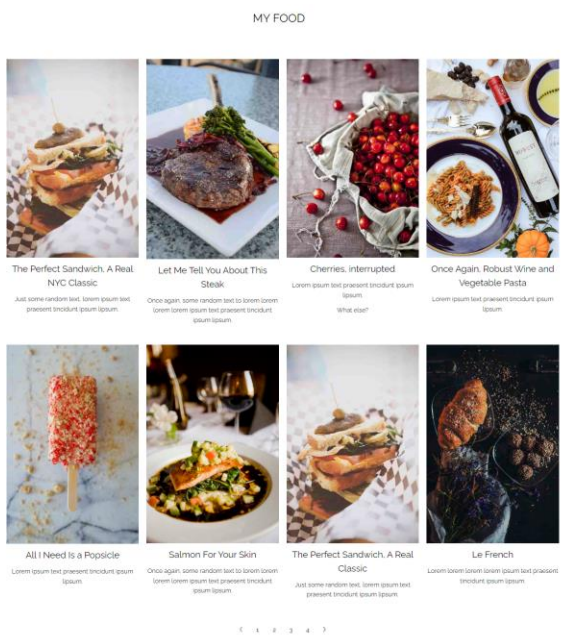
2. 레이아웃 상세 (메인)



메인 – section 2

- 특이사항 없이 block 정렬
- image의 경우 기본적으로 브라우저 사이즈에 맞게 조절되지만, 본래 이미지 크기 보다는 커지지 않도록 처리 필요

3. 구현 결과



- 브라우저 width 600px 이하에서 block 배열
- 미디어 쿼리 적용에 따라 이미지 배열이 흐트러지지 않고 배열되도록 구현

3. 구현 결과


헤더

☰

MY FOOD


Mail

메인 section1




The Perfect Sandwich, A Real NYC Classic

Just some random text, lorem ipsum text praesent tincidunt ipsum lipsum.



Let Me Tell You About This Steak


Once again, some random text to lorem lorem lorem lorem ipsum text praesent tincidunt ipsum lipsum.



Cherries, interrupted

Lorem ipsum text praesent tincidunt ipsum lipsum.

What else?




Once Again, Robust Wine and Vegetable Pasta

Lorem ipsum text praesent tincidunt ipsum lipsum.

메인 section2

About Me, The Food Man



I am Who I Am!

With Passion For Real, Good Food

Just me, myself and I, exploring the universe of unknowmment. I have a heart of love and an interest of lorem ipsum and mauris neque quam blog. I want to share my world with you. Praesent tincidunt sed tellus ut rutrum. Sed vitae justo condimentum.


푸터

FOOTER

Praesent tincidunt sed tellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla.

Powered by [w3.css](#)

BLOG POSTS



Lorem
Sed mattis nunc



Ipsum
Praes tinci sed

POPULAR TAGS

- Travel New York Dinner Salmon
- France Drinks Ideas Flavors
- Cuisine Chicken Dressing Fried
- Fish Duck

3. 구현 결과 - 전체 영역

CSS

```
/* ----- 전체 영역 */
* {
  box-sizing: border-box;
}

html,
body {
  width: 100%;
}

body {
  font-family: 'Raleway', sans-serif;
  max-width: 1220px;
  margin: 0 auto;
  position: relative;
}
```

```
a {
  all: unset;
  cursor: pointer;
}

hr {
  border-top: 1px solid #ccc;
}

h2 {
  font-weight: normal;
}

.inner {
  margin: 30px 0;
  padding: 5px 25px;
}
```

- CSS reset 파일 활용하여 초기화
- 이후 문서 전체에 적용할 CSS 추가 작성
 - 웹폰트, 전체 width, margin 등
- 전체를 감싸는 margin, padding 값을 통일시키기 위해 전용 클래스(.inner)를 만들어 모든 구역에 일정한 값이 적용되도록 작성

3. 구현 결과 - 헤더

HTML

```
<!-- 헤더 -->
<header>
  <div class="head-container">
    <div class="head-inner">
      <div class="nav-image">
        <a href="#"></a>
      </div>
      <h1><a href="#">MY FOOD</a></h1>
      <h2><a href="#">Mail</a></h2>
    </div>
  </div>
</header>
```

header > (container) > img

h1

h2



- 전체 영역을 header, main, footer 태그로 구분
- 시멘틱 고려하여 작성
 - 좌측 메뉴 <nav> 태그
 - 중간 로고 <h1> 태그
 - 우측 메뉴 <h2> 태그
- 좌측 메뉴는 JS 구현을 대체하여 하이퍼링크 적용
 - 픽토그램 다운로드하여 이미지 삽입함

3. 구현 결과 - 헤더

CSS

```
/* ----- 헤더 영역 */
header {
  width: 100%;
  background-color: #fff;
  position: fixed;
  top: 0;
  left: 0;
  z-index: 1000;
}

header .head-inner {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

header .nav-image {
  padding: 20px;
}
```

```
header .nav-image:hover {
  background-color: #ccc;
}

header .head-inner img {
  width: 15px;
  height: 15px;
  vertical-align: middle;
}

header .head-inner h1 {
  font-size: 24px;
  font-weight: normal;
}

header .head-inner h2 {
  font-size: 16px;
  padding: 0 20px;
}
```



- 추후 유지보수 가능성을 고려하여 최대한 쉽게 알아볼 수 있도록 코드를 표기하고자 함
- 전체 메뉴는 각각 block 레벨 요소로 구성한 후 display: flex; 로 수평 정렬
- 좌측 메뉴에 hover 효과 적용

3. 구현 결과 - 메인 (section1)

HTML

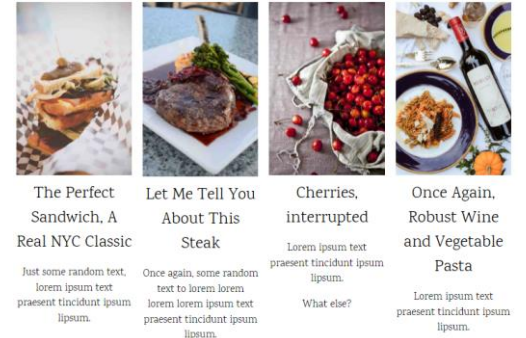
```
<!-- 메인 -->
<main>
  <!-- 메뉴 인사이트 -->
  <section>
    <div class="inner">
      <article class="insight-container">
        <div class="insight-item">
          <div class="item-img">
            
          </div>
          <h3 class="item-title">
            The Perfect Sandwich, A Real NYC Classic
          </h3>
          <p class="item-text">
            Just some random text, lorem ipsum text praesent tincidunt
          </p>
        </div>
      </article>
    </div>
  </section>
</main>
```

main > section > (.container) > (.item) > img

h3

p

```
<nav class="article-pages">
  <p> << >> </p>
  <p>1</p>
  <p>2</p>
  <p>3</p>
  <p>4</p>
  <p>>> </p>
</nav>
```



- 지난 과제에서는 모든 item을 수평 정렬한 것이 문제
 - 미디어 쿼리 적용 시 item이 의도한 4열이 아닌 3열로 배열되거나 여백이 지나치게 뜨는 등의 문제 발생
- 이번 과제에서는 item을 4개씩 한 번 더 container로 묶어줌
 - 미디어 쿼리 조건에 따라 해당 container가 배열 되도록 조치 → 이상한 배열 없이 4열로 고정
 - width:25% 적용함 → 여백이 뜨거나 item이 한쪽으로 쏠리지 않고 균등하게 적용
- container 내에는 각각 item 요소를 따로 넣고 안에 이미지, 타이틀 등 내용을 담아줌
- 하단 페이지는 nav 태그를 활용하여 구성함

3. 구현 결과 - 메인 (section1)

CSS

```
/* ----- 메인 영역 */
main {
  position: relative;
  top: 80px;
}

main .inner {
  margin: 30px 0;
  padding: 5px 25px;
}

/* 아티클 페이지 */
main nav {
  display: flex;
  justify-content: center;
  align-content: center;
}

main nav p {
  width: 35px;
  height: 35px;
  font-size: 0.8rem;
  line-height: 1.3rem;
  text-align: center;
}

main nav p:hover {
  background-color: #000;
  color: #fff;
}
```

```
/* 아티클 */
main .insight-item {
  padding: 0 8px;
}

main .item-img img {
  max-width: 100%;
  min-width: 100%;
}

main .insight-item {
  text-align: center;
  margin-bottom: 30px;
}

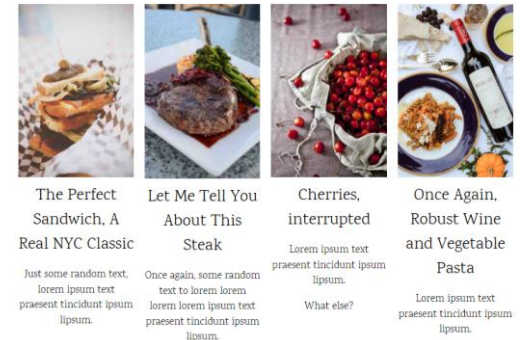
main .insight-item .item-title {
  font-weight: normal;
  line-height: 1.8rem;
  margin: 5px 0;
}

main .insight-item .item-text {
  font-size: 0.8rem;
  line-height: 1.3rem;
  margin: 10px 0;
}
```



▲ 아티클 페이지

▶ 아티클



- 상단 header에 position이 적용되어 main이 말려 올라가므로 해당 부분만큼 top 적용하여 하단으로 위치 조정
- item 이미지는 기본적으로 브라우저 크기에 맞게 조정되므로 max/min-width 적용
- 문장 간 가독성을 위한 간격 조정 / item에 margin 적용
- 페이지 선택 부분에 hover 효과 적용

3. 구현 결과 - 메인 (section2)

HTML

```
<!-- 셰프소개 -->
<section>
  <div class="inner">
    <article class="chef-container">
      <div class="chef-title">
        <h2>About Me, The Food Man</h2>
      </div>
      <div class="chef-image">
        
      </div>
      <div class="chef-introduce">
        <h3>I am Who I Am!</h3>
        <h4>With Passion For Real, Good Food</h4>
        <p>Just me, myself and I, exploring the universe of unknownme
      </div>
    </article>
  </div>
</section>
<div class="inner pagecut">
  <hr>
</div>
```

main > section > (.container) > title

image

h3

h4

p

About Me, The Food Man



I am Who I Am!

With Passion For Real, Good Food

Just me, myself and I, exploring the universe of unknownment. I have a heart of love and an interest of
lorem ipsum and mauris neque quam blog. I want to share my world with you. Praesent tincidunt sed

- 이후 CSS 적용할 것을 고려하여 각 글자, 이미지 간 구분하여 작성

3. 구현 결과 - 메인 (section2)

CSS

```
/* 셰프 소개 */
main .chef-container {
  text-align: center;
  margin-bottom: 40px;
}

main .chef-image img {
  max-width: 100%;
  margin-bottom: 15px;
}

main .chef-introduce h4 {
  font-weight: normal;
  font-style: italic;
}

main .chef-introduce p {
  font-size: 0.8rem;
  line-height: 1.3rem;
}
```

- 전체 내용 중앙 정렬
- 가독성을 위한 문장 행간 등 설정

About Me, The Food Man



I am Who I Am!

With Passion For Real, Good Food

Just me, myself and I, exploring the universe of unknownment. I have a heart of love and an interest of lorem ipsum and mauris neque quam blog. I want to share my world with you. Praesent tincidunt sed

3. 구현 결과 - 푸터

HTML

```
<!-- 푸터 -->
<footer>
  <div class="inner">
    <div class="f-container footer">
      <h2>FOOTER</h2>
      <p>Praesent tincidunt sed tellus ut rutrum. Sed vitae
      <p>Powered by <a href="">w3.css</a></p>
    </div>
    <div class="f-container blog">
      <h2>BLOG POSTS</h2>
      <div class="blog-posts">
        <div class="post-image">
          
        </div>
        <div class="post-textbox">
          <h3 class="post-title">Lorem</h3>
          <p class="post-text">Sed mattis nunc</p>
        </div>
      </div>
    </div>
    <div class="f-container tag">
      <h2>POPULAR TAGS</h2>
      <div class="tag-box">
        <span class="tag-item">Travel</span>
        <span class="tag-item">New York</span>
      </div>
    </div>
  </div>
</footer>
```

footer > (.footer) > 각 text

(.blog) > (.blog-post) > 각 image, text

(.tag) > 각 text 및 tag들

FOOTER

Praesent tincidunt sed tellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla.

Powered by [w3.css](#)

BLOG POSTS



Lorem
Sed mattis nunc



Ipsium
Praes tinci sed

POPULAR TAGS

Travel New York Dinner
Salmon France Drinks
Ideas Flavors Cuisine
Chicken Dressing Fried
Fish Duck

- 〈FOOTER〉, 〈BLOG POST〉, 〈POPULAR TAGS〉를 각각 하나의 영역으로 구분하여 작성
- 〈FOOTER〉 영역에 일부 하이퍼링크 작성
- 〈BLOG POSTS〉 영역의 경우 내용물을 따로 div 태그에 담아 두 묶음으로 작성

3. 구현 결과 - 푸터

CSS

```
/* ----- 푸터 영역 */
footer {
  position: relative;
  top: 80px;
}

footer .f-container {
  margin-bottom: 30px;
}

footer p {
  font-size: 0.8rem;
  line-height: 1.3rem;
}

footer a {
  text-decoration: underline;
}

/* 푸터-태그 */
footer .tag .tag-item {
  background-color: #ccc;
  color: black;
  padding: 2px 5px;
  font-size: 0.8rem;
  line-height: 2rem;
}

footer .tag .tag-item:first-child {
  background-color: #000;
  color: #fff;
  font-size: 0.9rem;
}

/* 푸터-블로그 */
footer .blog {
  display: flex;
  flex-direction: column;
}

footer .blog .blog-posts {
  padding: 10px 5px;
  border-bottom: 1px solid #f1f1f1;
  display: flex;
  align-items: center;
}

footer .blog .blog-posts:hover {
  background-color: #f1f1f1;
}

footer .blog .blog-posts:last-child {
  border: none;
}

footer .blog .post-image img {
  width: 50px;
  height: 50px;
}

footer .blog .post-textbox {
  margin: 0 0 20px;
}

footer .blog .post-textbox .post-title {
  font-weight: normal;
  margin: 0;
}

footer .blog .post-textbox .post-text {
  margin: 0;
}
```

FOOTER

Praesent tincidunt sed tellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla.

Powered by [w3.css](#)

BLOG POSTS



Lorem
Sed mattis nunc



Ipsium
Praes tinci sed

POPULAR TAGS

Travel New York Dinner
Salmon France Drinks
Ideas Flavors Cuisine
Chicken Dressing Fried
Fish Duck

- 전체 footer 영역에 미디어 쿼리에 따라 배열이 바뀌어도 가독성 문제 없도록 margin값 등 적용함
- 〈BLOG POSTS〉 영역은 item(포스트 내용 묶음)을 flex로 수직 배열하고, 각각의 item 내에서 한 번 더 flex로 수평 정렬 적용
- tag item은 first-child만 따로 선택하여 검은색 강조

4. 소요 시간 및 후기

소요시간

- 코딩 4:00~ 6:30 (2시간 30분)

후기

- 지난 번 과제에서 미디어 쿼리 구현에 실패했기 때문에 (배열이 뒤트림) 이번에는 그 부분을 꼭 해결하고 싶었습니다.
- 지난 과제 페이지 소스를 뜯어보아도 명쾌한 방법이 안 보여서 답이 없는 채로 무작정 코딩을 시작했는데, 중간에 container 배치를 떠올리고 적용해본 결과 문제없이 작동하는 걸 확인했습니다.
- 따라서 이번 구현은 미디어 쿼리까지 원본 페이지와 완전히 똑같이 작동합니다. 너무너무 뿌듯했고, 의욕도 충만해지는 시간이었습니다!