



클론 코딩(4일차)

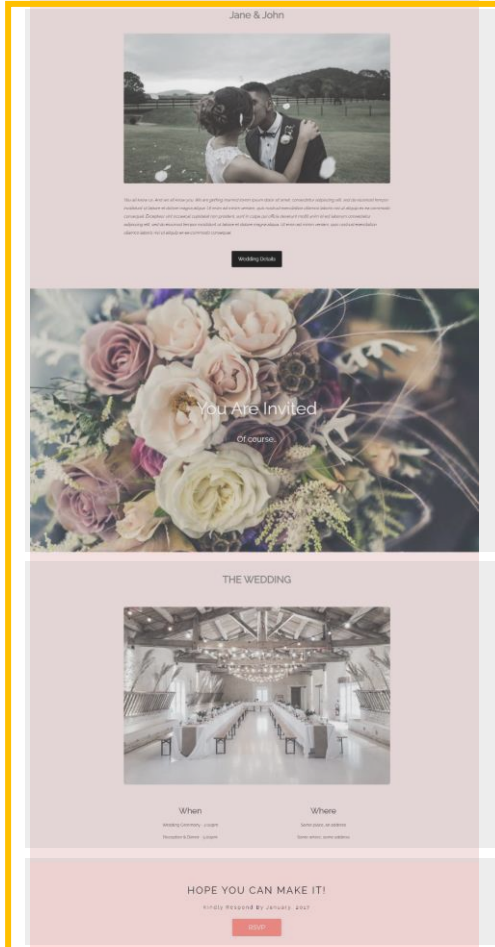
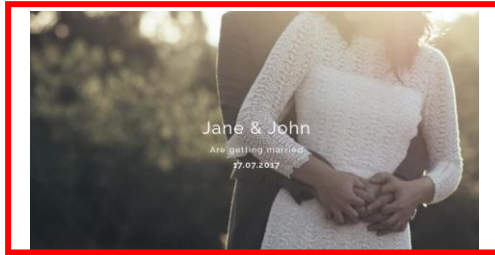
풀스택 자바 웹개발자 오후반
방미소



목차

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

1. 레이아웃 정의 (전체)



Section1

Section2

Section3

헤더

- 브라우저 height에 꼭 맞는 큰 이미지 삽입

네비게이션

- 브라우저를 스크롤해도 하단에 고정 / 반투명

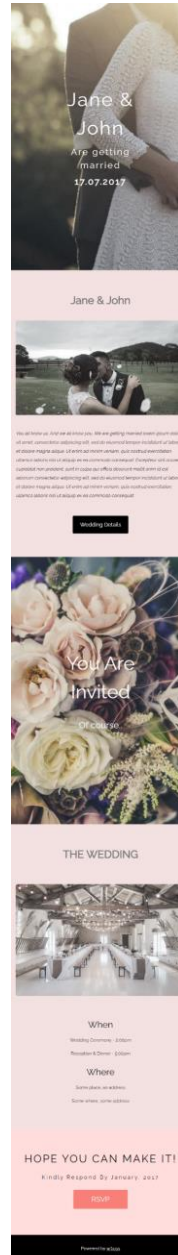
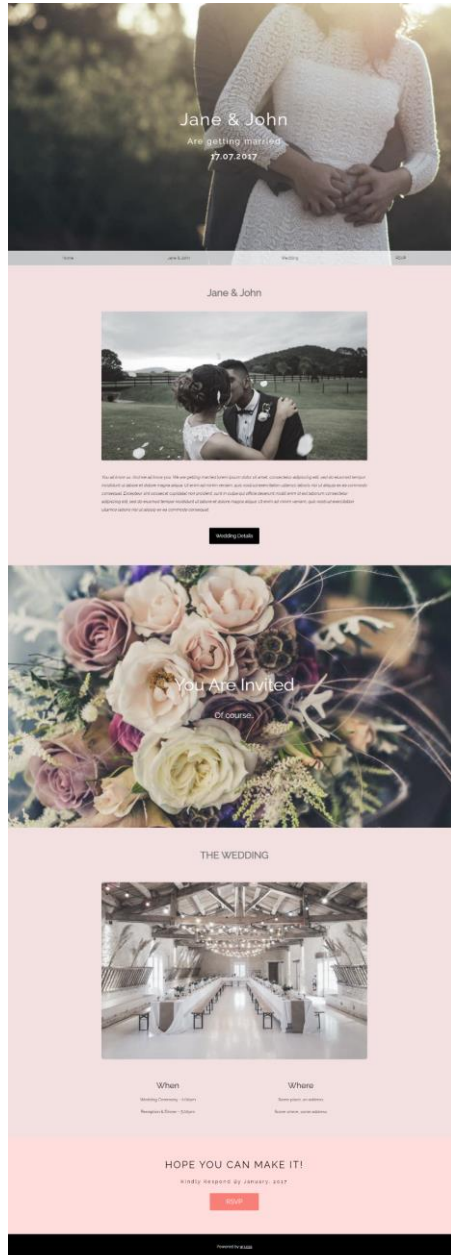
메인

- 크게 3개의 section으로 구분

푸터

- 특이사항 없음

1. 레이아웃 정의 (미디어쿼리)

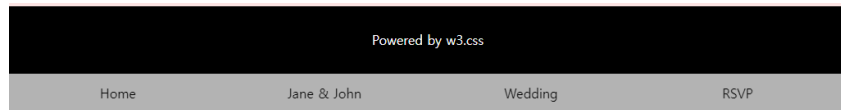
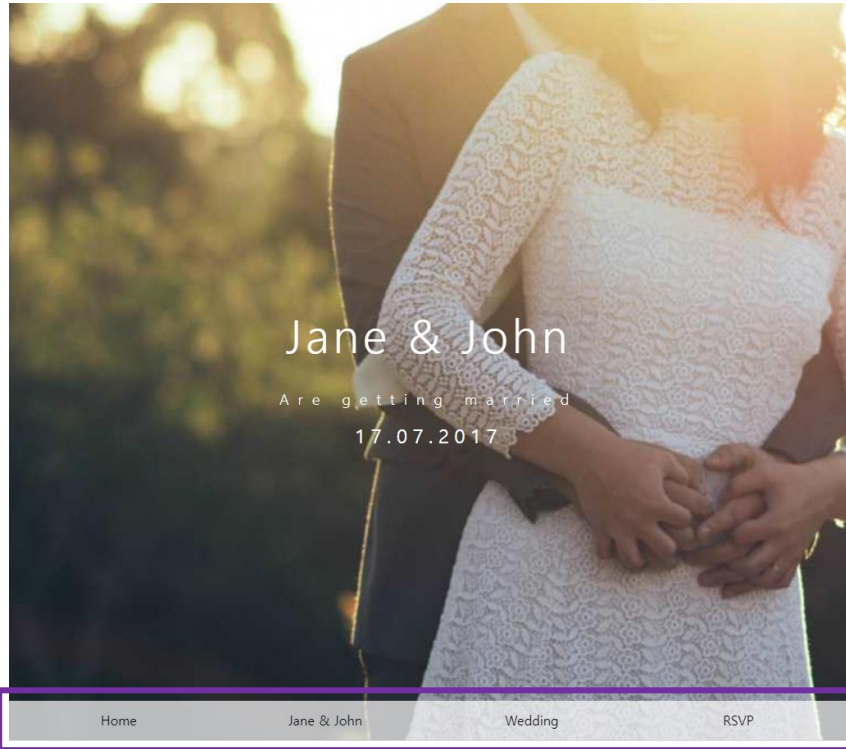


미디어 쿼리

0~600px

> 하단 메뉴가 사라짐

2. 레이아웃 상세 (헤더, 푸터)



헤더



네비게이션

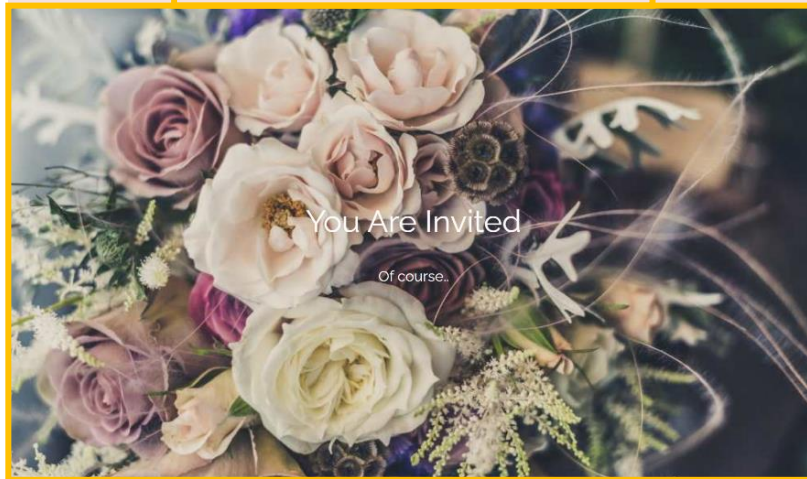
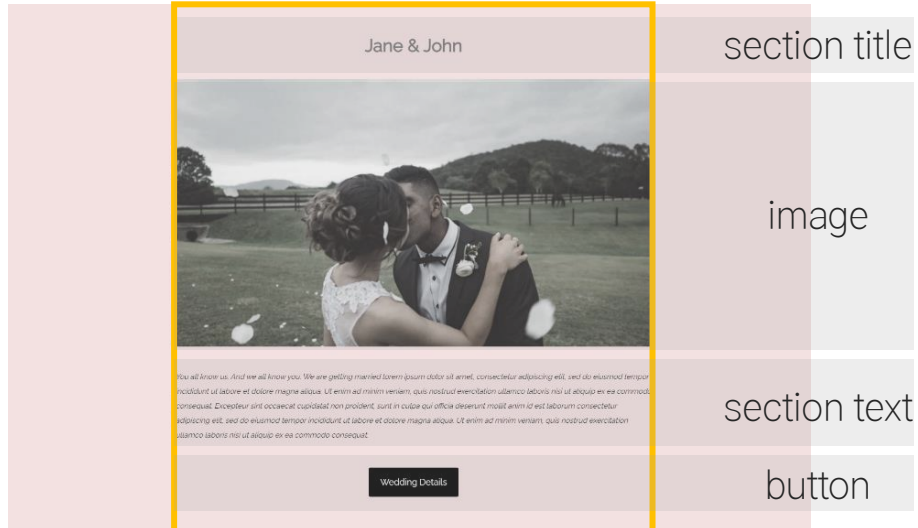
- 상단 타이틀 영역
 - 큰 이미지로 디자인 효과
 - 글자가 이미지 상단에 위치함
- 메뉴바는 브라우저를 스크롤해도 하단에 고정 (fixed)
 - 기본적으로 반투명
 - hover시 완전 불투명 + 버튼에 회색 or 검은색 컬러 효과



푸터

- 특이사항 없음

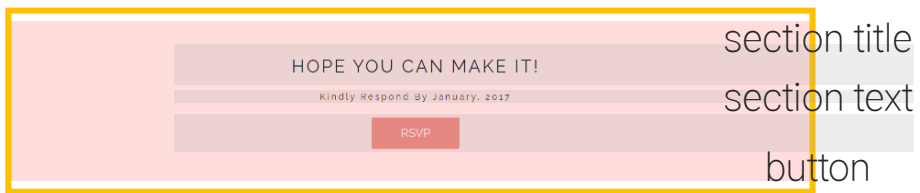
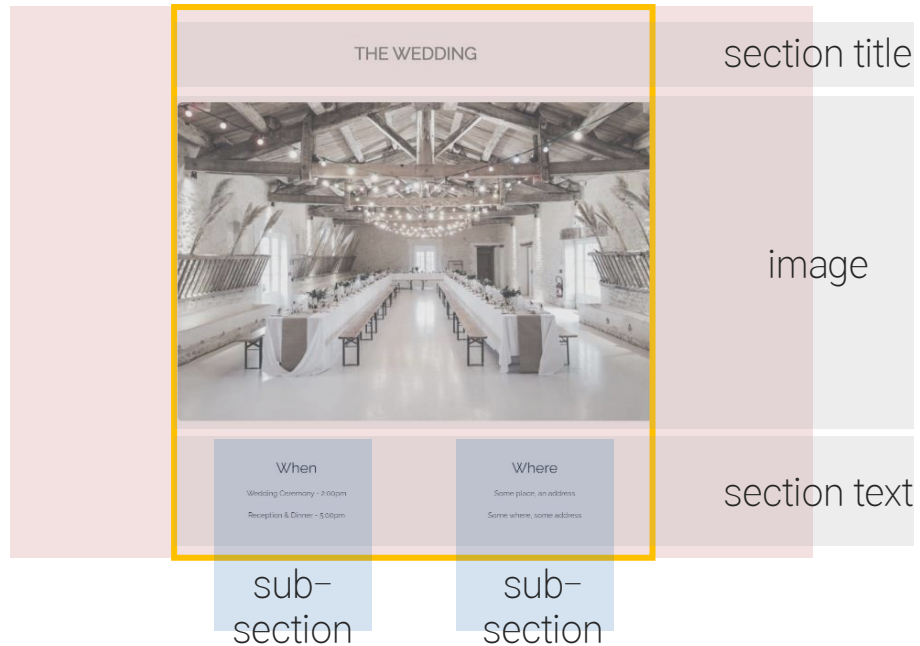
2. 레이아웃 상세 (메인)



메인 – section 1

- 하단 이미지를 별도의 section으로 작성할 수도 있겠으나 그만큼의 정보성이 없는 것 같아 위 영역에 포함시킴
- 각각 width가 다르게 적용되어야 하므로 상하로 분리

2. 레이아웃 상세 (메인)



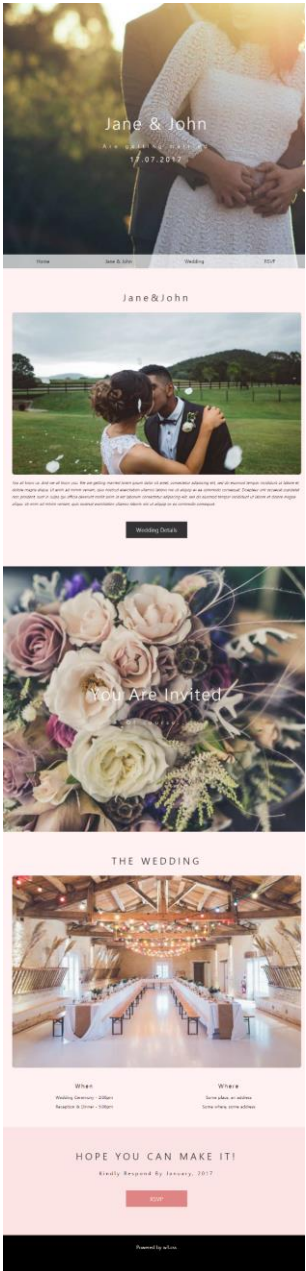
메인 – section 2

- 보이는 부분대로 분리

메인 – section 3

- 배경 색이 살짝 다름
- button 크기와 모양은 section1의 button과 동일

3. 구현 결과



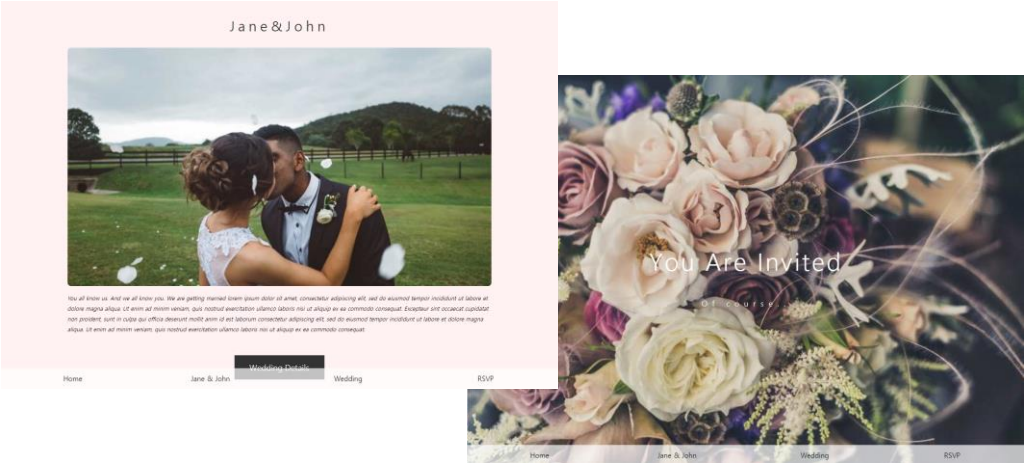
- 브라우저 width 600px 이하에서 네비게이션 보이지 않음

3. 구현 결과

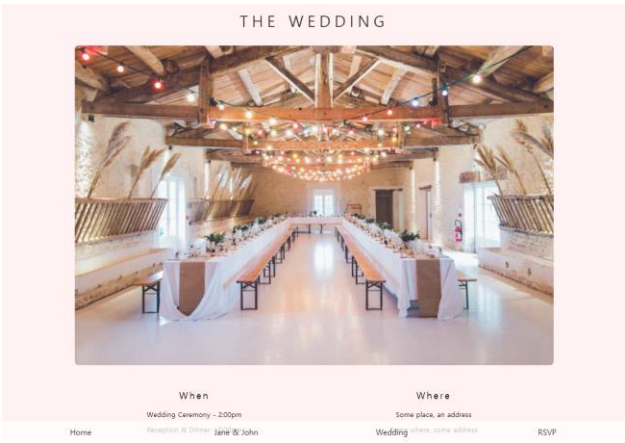
헤더



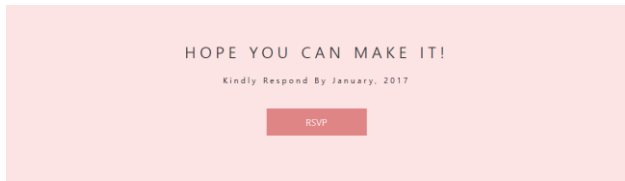
메인 section1



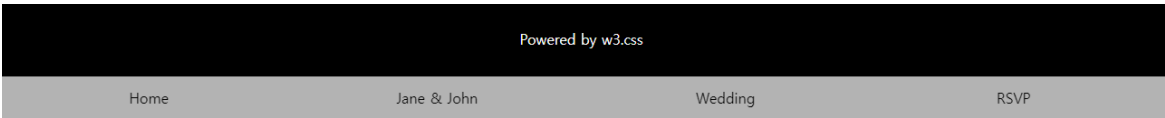
메인 section2



메인 section3



푸터



3. 구현 결과 - 전체 영역

CSS

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

body {
  background-color: #fff2f2;
  margin: 0 auto;
}

a {
  all: unset;
  cursor: pointer;
}

ul,
ol,
li {
  padding: 0;
  margin: 0;
  list-style: none;
}

button {
  all: unset;
}
```

```
.invisible-content {
  display: none;
}

.inner {
  margin: 0 1.5rem;
  padding: 3rem 0;
}

.outer {
  height: 100vh;
  position: relative;
}

.outer-title {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  color: #fff;
  text-align: center;
}

.outer-title.big-title {
  font-weight: 300;
  font-size: 3.5rem;
  letter-spacing: 0.3rem;
}
```

```
.button-container {
  margin: 3rem 0;
  text-align: center;
}

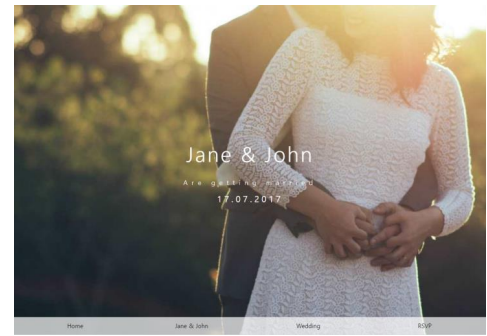
.center-button {
  width: 13rem;
  height: 3.5rem;
  line-height: 3.5rem;
  font-size: 1.2rem;
  font-weight: 300;
  display: inline-block;
  color: #fff;
}

.box-round {
  border-radius: 0.5rem;
}
```

- CSS reset 파일 활용하여 초기화
- 이후 문서 전체에 적용할 CSS 추가 작성
 - 웹폰트, 전체 width, margin 등
- 안으로 들어간 section에는 .inner 클래스, 브라우저에
꼭차는 큰 이미지의 경우 container에 .outer 클래스
적용하여 margin 및 padding 값 통일

3. 구현 결과 - 헤더, nav

HTML



```
<header>
  <div class="outer">
    <div class="outer-title">
      <h1 class="big-title">Jane & John</h1>
      <p class="big-subtitle">Are getting married</p>
      <h2>17.07.2017</h2>
    </div>
  </div>
</header>
```

header › (container) › h1
p
h2

```
<nav class="under-menu">
  <ul class="menu-container">
    <li><a href="#">Home</a></li>
    <li><a href="#jane-john">Jane & John</a></li>
    <li><a href="#wedding">Wedding</a></li>
    <li><a href="#rsvp" class="menu-rsvp">RSVP</a></li>
  </ul>
</nav>
```

nav › ul › li

- 중요해 보이는 문구 순대로 h 태그 적용
- 이미지는 background로 적용할 것이므로 html에 삽입하지 않음
- 공통 margin/padding 적용할 .outer 컨테이너에 담아줌

- 목록 형태로 작성 후 메뉴 클릭 시 해당 section으로 이동하도록 조치

3. 구현 결과 - 헤더

CSS

```
/* ----- 헤더 & 큰이미지 */
header .outer {
  background: url(../img/wedding_couple.jpg) no-repeat;
  background-position: center;
  background-size: cover;
}

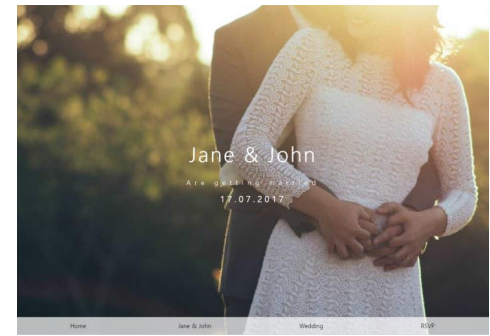
header .outer-title h2 {
  font-weight: 500;
  letter-spacing: 0.4rem;
}

main .outer {
  background: url(../img/flowers.jpg) no-repeat;
  background-position: center;
  background-size: cover;
}
```

```
.outer {
  height: 100vh;
  position: relative;
}

.outer-title {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  color: #fff;
  text-align: center;
}

.outer-title .big-title {
  font-weight: 300;
  font-size: 3.5rem;
  letter-spacing: 0.3rem;
}
```



- 타이틀을 포함하여 본문영역 큰 이미지까지 .outer 컨테이너 적용
- 텍스트 부분은 position:absolute; 적용한 후 transform:translate()으로 페이지 중앙에 이동

3. 구현 결과 - nav

CSS

```
/* ----- 메뉴 */
nav.under-menu {
  position: fixed;
  bottom: 0;
  left: 0;
  width: 100%;
  background-color: rgba(255, 255, 255, 0.7);
  z-index: 1000;
}

nav.under-menu:hover {
  background-color: rgba(255, 255, 255, 1);
}

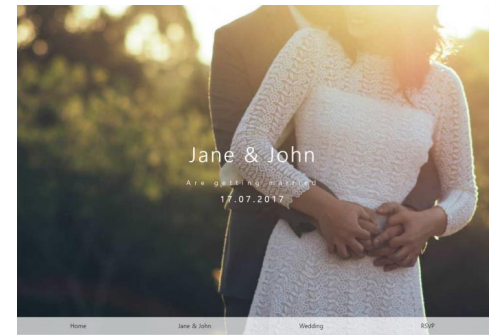
nav .menu-container {
  height: 3rem;
  padding: 0.5rem;
  display: flex;
  align-items: center;
}

nav .menu-container li {
  width: 25%;
}

nav .menu-container a {
  width: 100%;
  padding: 0.5rem 0;
  display: block;
  text-align: center;
}

nav .menu-container a:hover {
  background-color: #ccc;
}

nav .menu-container .menu-rsvp:hover {
  background-color: #000;
  color: #fff;
}
```



- 메뉴는 position:fixed 적용 후 하단에 배치
- 배경컬러에 투명도 적용하여 반투명하게 구현
- ul>li 로 작성된 메뉴를 display:flex로 수평 배열
- 이후 hover 적용

3. 구현 결과 - 메인 (section1)

HTML

```
<section id="jane-john">
  <h2 class="invisible-content">invite you!</h2>
  <div class="inner">
    <div class="section-item-container">
      <article>
        <p class="section-title">Jane&John</p>
        <div class="section-image">
          
        </div>
        <p class="text">
          You all know us. And we all know you. We are getting married
          lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
          eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
          enim ad minim veniam, quis nostrud exercitation ullamco laboris
          nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat
          cupidatat non proident, sunt in culpa qui officia deserunt
          mollit anim id est laborum consectetur adipiscing elit, sed do
          eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
          enim ad minim veniam, quis nostrud exercitation ullamco laboris
          nisi ut aliquip ex ea commodo consequat.
        </p>
        <div class="button-container">
          <a href="#Wedding" class="center-button">Wedding Details</a>
        </div>
      </article>
    </div>
  </div>
</div>
<div class="outer">
  <div class="outer-title">
    <p class="big-title">You Are Invited</p>
    <p class="big-subtitle">Of course..</p>
  </div>
</div>
</section>
```

main > section > (container) > article > p, image, ...
(container) > image



- width를 좁게 적용할 상단부분과 넓게 적용할 하단부분을 구분하여 각각 .inner, .outer 클래스 적용 -> CSS
- 각 section에 id 지정한 후 메뉴에 하이퍼링크 적용
- 버튼은 a 태그 사용

3. 구현 결과 - 메인 (section1)

CSS

```
/* ----- 메인 */
section .section-item-container {
  max-width: 980px;
  margin: 0 auto;
}

.section-title {
  color: #333;
  font-size: 2rem;
  letter-spacing: 0.5rem;
  text-align: center;
}

.section-image {
  max-width: 100%;
  text-align: center;
}

.section-image img {
  max-width: 100%;
  width: 100%;
  object-fit: contain;
}

/* 제인&존 결혼 */
#jane-john article .text {
  font-size: 0.8rem;
  font-style: italic;
  line-height: 1.5rem;
}

#jane-john .center-button {
  background: #333;
}
```



- 각 영역 서식에 맞게 지정
- 하단의 큰 이미지는 header 영역 css를 공통으로 적용

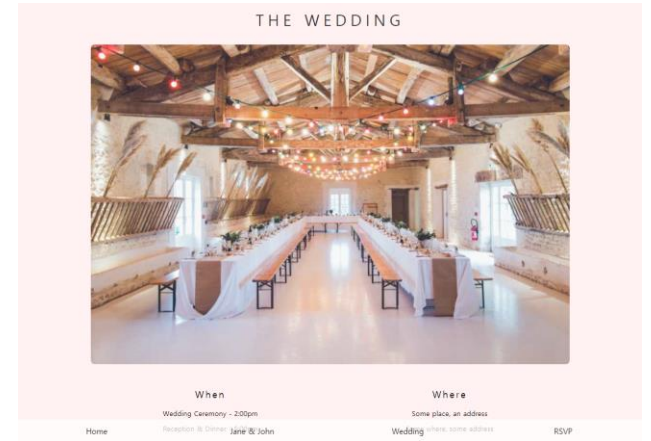
3. 구현 결과 - 메인 (section2)

HTML

```
<section id="wedding">
  <h2 class="invisible-content">the information</h2>
  <div class="inner">
    <div class="section-item-container">
      <article>
        <p class="section-title">THE WEDDING</p>
        <div class="section-image">
          
        </div>
      </article>
      <article class="wedding-section-container">
        <section class="sub-section">
          <h3>When</h3>
          <p>Wedding Ceremony - 2:00pm</p>
          <p>Reception & Dinner - 5:00pm</p>
        </section>
        <section class="sub-section">
          <h3>Where</h3>
          <p>Some place, an address</p>
          <p>Some where, some address</p>
        </section>
      </article>
    </div>
  </div>
</section>
```

main › section › (.container) › article › p, image

article › subsection, subsection



- 하단에 subsection 을 2개 만들어 각각 컨텐츠 작성함

3. 구현 결과 - 메인 (section2)

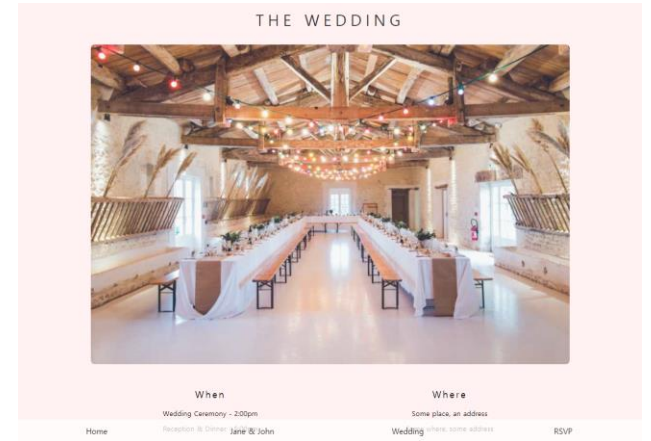
CSS

```
/* 인포메이션 */
#wedding .wedding-section-container {
  display: flex;
}

#wedding .sub-section {
  width: 50%;
  text-align: center;
}

#wedding .sub-section h3 {
  font-weight: normal;
  letter-spacing: 0.2rem;
  margin-top: 3rem;
}

#wedding .sub-section p {
  font-size: 0.9rem;
}
```



- subsection 컨테이너에 flex 적용하여 하단 subsection 수평 배열

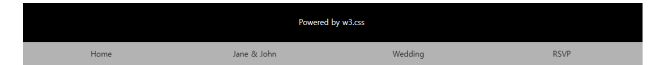
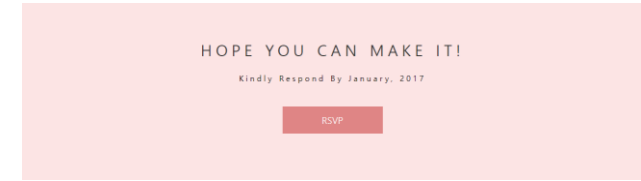
3. 구현 결과 - 메인 (section3, 푸터)

HTML

```
<section id="rsvp">
  <div class="inner">
    <h2 class="invisible-content">RSVP</h2>
    <p class="section-title">HOPE YOU CAN MAKE IT!</p>
    <p>Kindly Respond By January, 2017</p>
    <div class="button-container">
      <button class="center-button">RSVP</button>
    </div>
  </div>
</section>
```

main › section › (.container) › h2, p, button, ...

```
<footer>
  <p>Powered by w3.css</p>
</footer>
```



- 클리 및 button 작성

- 특이사항 없이 내용 작성

3. 구현 결과 - 메인 (section3, 푸터)

CSS

```
/* rsvp */
#rsvp {
  background-color: ■■rgb(252, 228, 228);
  text-align: center;
}

#rsvp p.section-title ~ p {
  letter-spacing: 0.3rem;
}

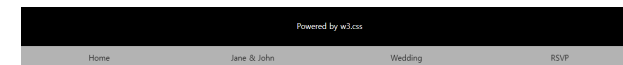
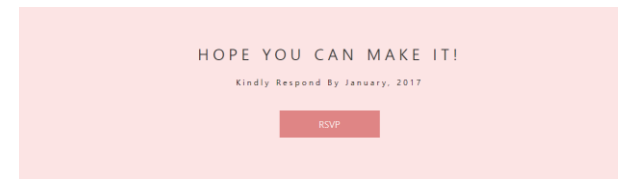
#rsvp .center-button {
  background: ■■rgb(223, 133, 133);
}
```

- 해당영역은 위와 달리 배경색이 다르므로 배경색 적용
- 버튼 크기나 정렬 등은 상단 태그로 설정되었기 때문에 색상만 적용

```
/* ----- 푸터 */
footer {
  background-color: □■#000;
  padding: 1rem 0 4rem 0;
  color: ■■#fff;
  text-align: center;
}

/* ----- 미디어쿼리 */
@media only screen and (max-width: 600px) {
  nav.under-menu {
    display: none;
  }
}
```

- 푸터 배경색 및 서식 적용
- 미디어쿼리로 특정 브라우저 너비 이하에서 메뉴 사라지도록 적용



4. 소요 시간 및 후기

소요시간

- 코딩 : 2시간 30분

후기

- 페이지 구성 자체는 이전 과제들에 비해 상대적으로 간단했기 때문에 이번에는 CSS 작성 시 px 단위를 사용하지 않고 디바이스에 따라 동일하게 보이도록 최대한 상대 단위(rem)을 써보려고 시도했습니다.
- 다음 번에는 태그를 최대한 중복 없이 효율적으로 쓰는 데 초점을 두려고 합니다.