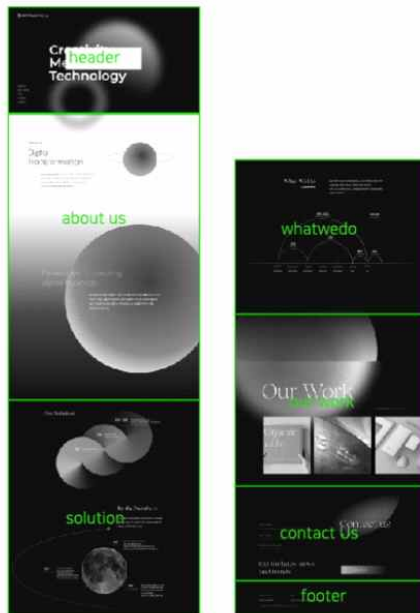


## 파일

main	images	favicon logo-white.png neo-logo.png ... (images files)
	styles	reset.css variable.css fonts.css pretendardvariable.css
	fonts	
		index-ah-oh.html
		index-m.css
		index-pc_j.css
		index-pc_K.css
		index-pc_M.css
		index-J.js
		index-M.js



## 퍼블리싱 가이드라인 (1)

### 전체 공통 가이드라인

#### 사이즈 (폰트 및 해상도)

- 모바일 기준으로 작업 진행. 모바일 완성 시, PC버전 진행.
- 모바일 기준: 380px (아이폰13pro)
- PC 기준: 1200px 작업.
- 브레이크 포인트(Media Query) : 640px
- 폰트 사이즈는 rem 기준.
- 이미지 사이즈는 % 기준.

#### 작성 표기

- 주석 2줄 작성: 큰 섹션 ":"=, 작은 섹션 ":"=
- 이미지 네이밍 규칙
- 영역 네이밍 규칙
- HTML, CSS에서 네이밍은 개발 표기법으로 작성.
- 최대한 클래스명으로 작업 진행할 수 있도록 하며, 최대한 아이디명은 사용 자제.
- 클래스명 생성은 섹션 네이밍 기준으로 작성하며, 충분히 해당 클래스의 기능을 확인할 수 있는 단어 사용.

```

urapper (x) container (x) bg (x) btn (x)
header-urapper (o) header-container (o) about1-bg (o) about1-btn (o)
  
```

- 코드 작성은 줄 바꿈으로 작성. (자동 정렬 사용)
- 속성 값 작성 시 ""큰따옴표 사용. 작은 따옴표 사용 금지.
- W3 유효성 검사는 작업 시 항상 진행

### HTML

- index.html 하나의 파일로 함께 사용.
- header, main, footer 영역으로 크게 구분 후, 작은 영역은 div로 설계 진행.
- 작업 시, SSR용 외부 링크로 작성 → 1개의 파일로 정리.
- 외부 링크 위치는 상단 head 영역에 작성.

### CSS

- 공통 디자인은 variable.css를 활용하여 작업 진행.
- 작성 순서:
  - 큰 영역 → 작은 영역 순으로 작성.
  - 타입선택자보다 클래스명을 활용하여 작성.

```

h1 { color: #fff; }
.logo { color: #fff; }
  
```

- 모든 작성은 소문자로 사용. (클래스명, 색상, 코드, 등)

```

ex )
color : #160F3E; (x)
color : #160F3e; (o)
color: #FFF; (x)
color: #fff; (o)
단, RGB 반박될 경우 3자리 16진수로 표기한다.
  
```

### JavaScript, jQuery

- 선택지명, 네이밍, 값명 표기법을 적용.
- 변수 선언 시, 지역변수로 작성. (const 우선 작성)
- 다른 값을 할당해야 할 경우에만 let으로 선언하며, 주석을 작성하여 내용을 기록해 팀원이 확인할 수 있도록 한다.
- 전역 변수 선언은 최대한 사용 자제.
- ~ \$(선택지이트먼트) 작성 시, "별칭 사용, 이 외 "큰따옴표"를 이용하여 작성.

### SSAR

- 개발 공통