

# HTML+CSS+JS 포트폴리오 실전 퍼블리싱(시즌3)

Create a great web Ui Design styling & Development  
with HTML+CSS+JS



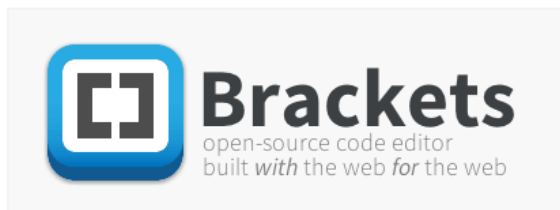
텍스트 에디터 사용법 및 웹 퍼블리싱 이론 및 이론 활용 예제

{ 웹 개발 이해하기(HTML, CSS, and JS) }

{ Microsoft Edge 피닉스 코드(Phoenix Code) }

☞ 텍스트 에디터가 라이브 미리보기(Live Preview) 기능과 코드 힌트(Code Hint)를 제공한다면,  
텍스트 에디터(Text Editor) 프로그램은 자신에게 맞는 에디터를 사용하면 됩니다.

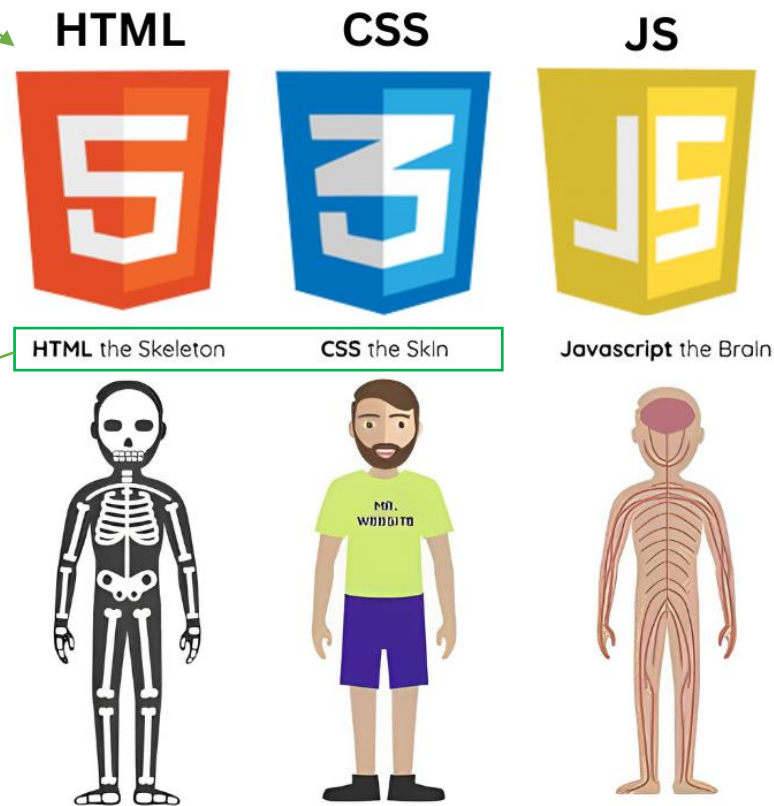
- Visual Studio Code(추천 프로그램)
- Atom
- Sublime Text
- EditPlus
- Notepad++
- Brackets(추천 프로그램)



Phoenix Code는  
기존 Brackets의  
업그레이드 버전

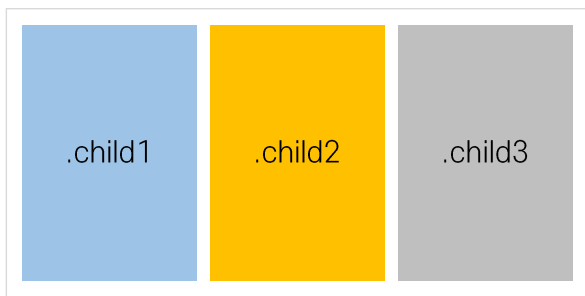


▲ 피닉스 코드 다운로드 : <https://phcode.io/>



▲ 이미지 출처 : <https://www.linkedin.com/pulse/building-blocks-web-development-html-css-javascript-sahil-kavitate>

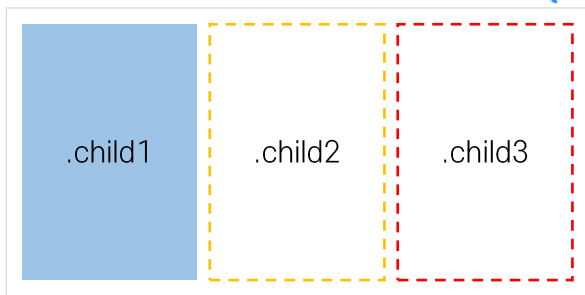
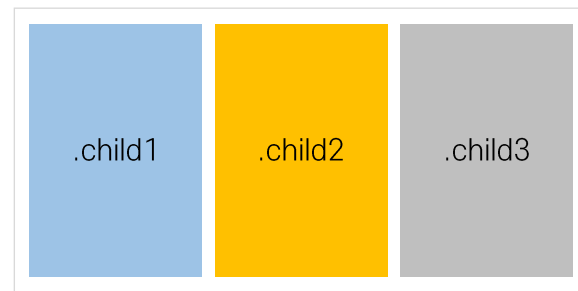
## { 요소를 안보이게 하는 3가지 방법(opacity, display, visibility) }



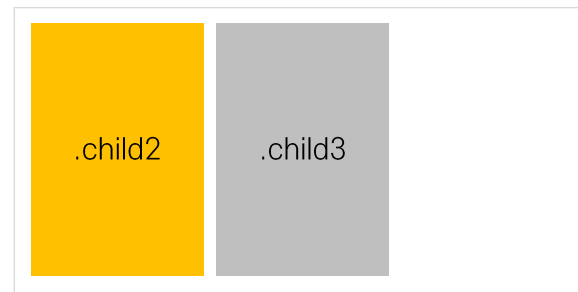
```
<div class="parent">  
  <div class="child1"></div>  
  <div class="child2"></div>  
  <div class="child3"></div>  
</div>
```

HTML

```
.child1 {  
  display: none;  
  /* 공간의 크기도 없어지면서 화면상 안  
  보이고 실제로도 존재하지 않음 */  
}
```



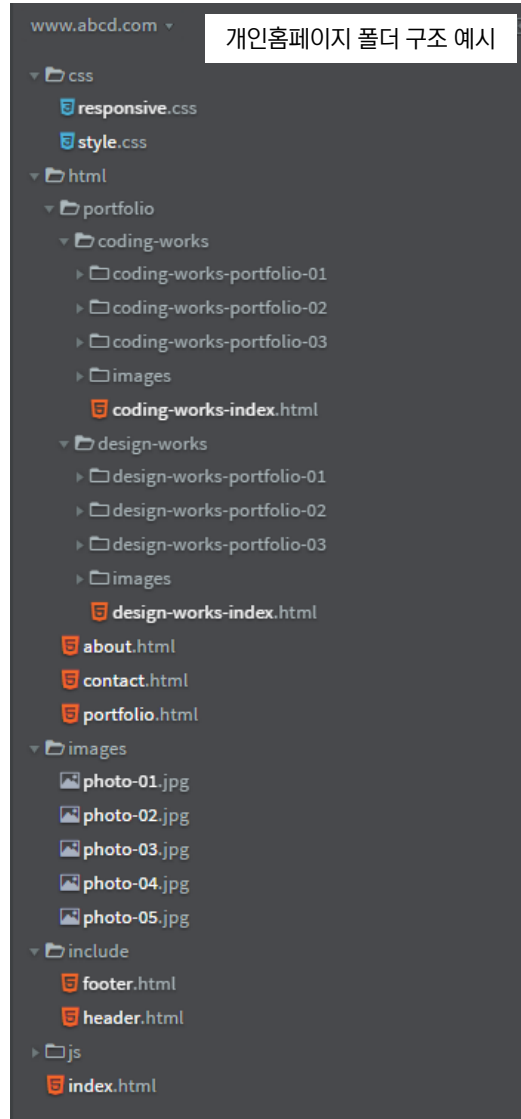
```
.child2 {  
  opacity: 0;  
  /* 공간의 크기는 유지하면서 화면상 안보이지만 실제로는 존재함 */  
}  
.child3 {  
  visibility: hidden;  
  /* 공간의 크기는 유지하면서 화면상 안보이면서 실제로는 존재하지도 않음 */  
}
```



HTML 태그와 요소(Element)의 차이?

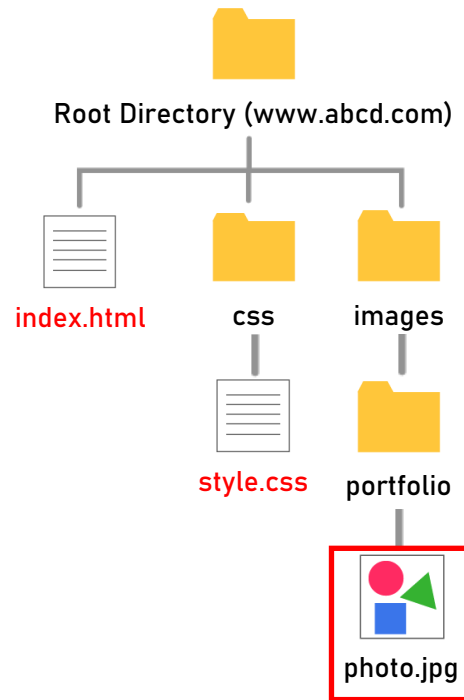


# [HTML+CSS+JS 포트폴리오 실전 퍼블리싱] 폴더 파일 경로(절대경로 vs 상대경로) 코딩웍스(Coding Works)



## 상대경로 or 상대주소

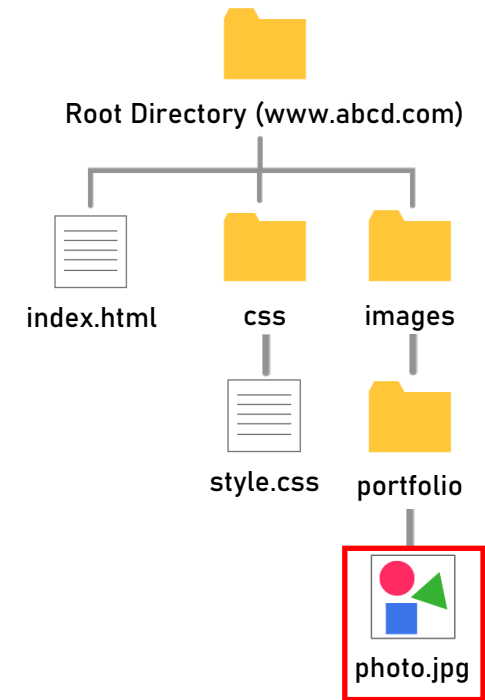
HTML 또는 CSS 파일을 기준으로 경로를 정하는 방식



Index.html 에서 링크하는 경우   
style.css 에서 링크하는 경우  
background-image: url(../images/portfolio/photo.jpg)

## 절대경로 or 절대주소

최상위 폴더(root)를 기준으로 경로를 정하는 방식



Index.html 에서 링크하는 경우   
style.css 에서 링크하는 경우  
background-image: url(/images/portfolio/photo.jpg)

▲ 코딩웍스 블로그 참고 : <https://www.inflern.com/blogs/1284>



## { 퍼블리싱의 모든 것! display 속성(인라인요소, 블록요소, 인라인블록) & 포지션 속성 }

**[필수]** 포지션 속성은 기본이 position: static 입니다. 곧, html이 나오는 순서에 맞춰 정적 (static) 으로 아래로 쌓이게 됩니다. 하지만 **요소를 원하는 위치에 놓고 싶을 경우** 포지션 속성을 사용한 부모요소와 자식요소의 관계 그리고 포지션 속성이 겹치는 경우 우선 순위를 기억해야 합니다.



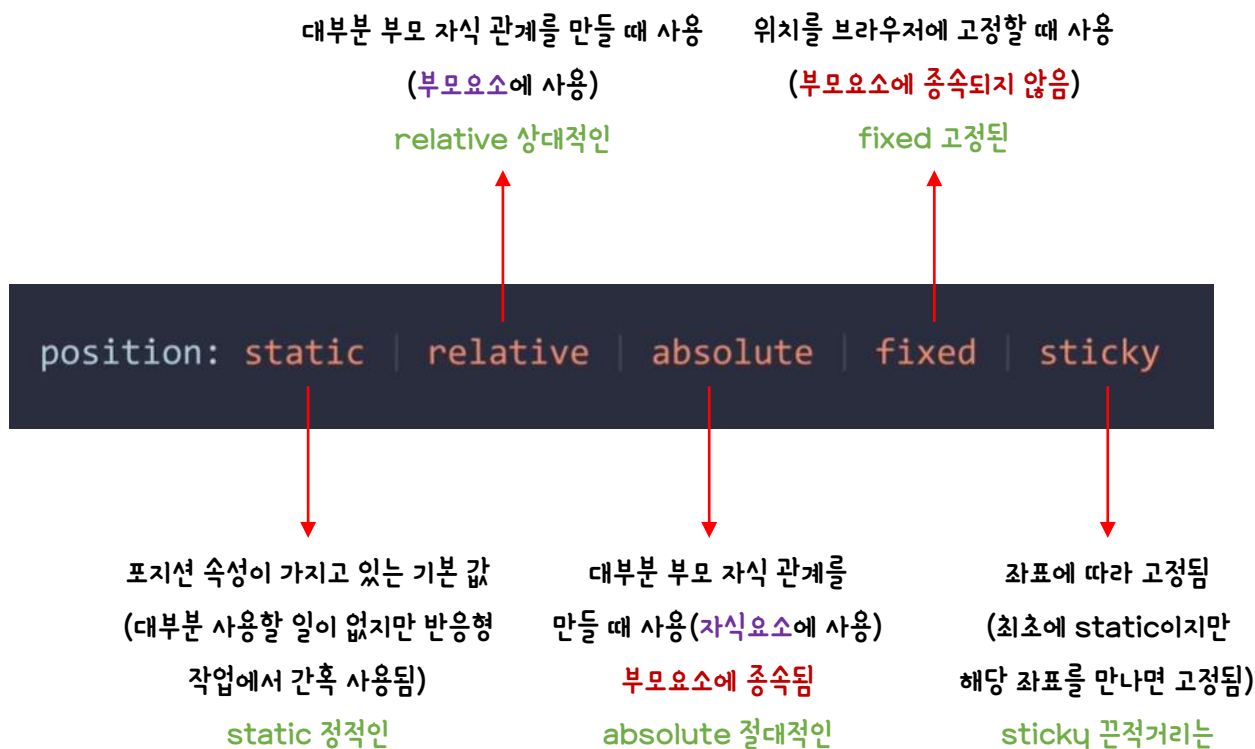
코딩웍스 유튜브 참고하기 - <https://youtu.be/qTgqyNRcos0?si=-RUmv1N1z4YVj73>

**[필수]** 퍼블리싱의 모든 것이라고 할 수 있는 **CSS의 display 속성** 그 중에서 **HTML 요소의 인라인요소, 블록요소, 인라인블록에 대한 특징**을 정확히 알고 있다면 앞으로 퍼블리싱이 수월해지실 것이고 아니라면 항상 애매한 상태에서 퍼블리싱 하시게 됩니다. **이건 너무 너무 중요합니다~!!!**



코딩웍스 블로그 참고하기 - <https://www.inflearn.com/blogs/376>

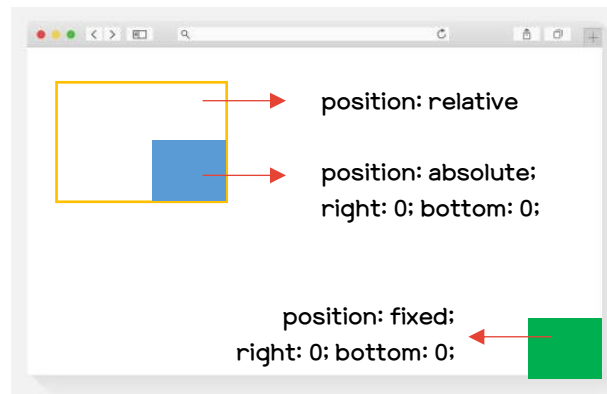
## { Position 속성의 완벽한 이해와 활용 #01 }



### 포지션 속성 사용할 때 기억해야 할 것들

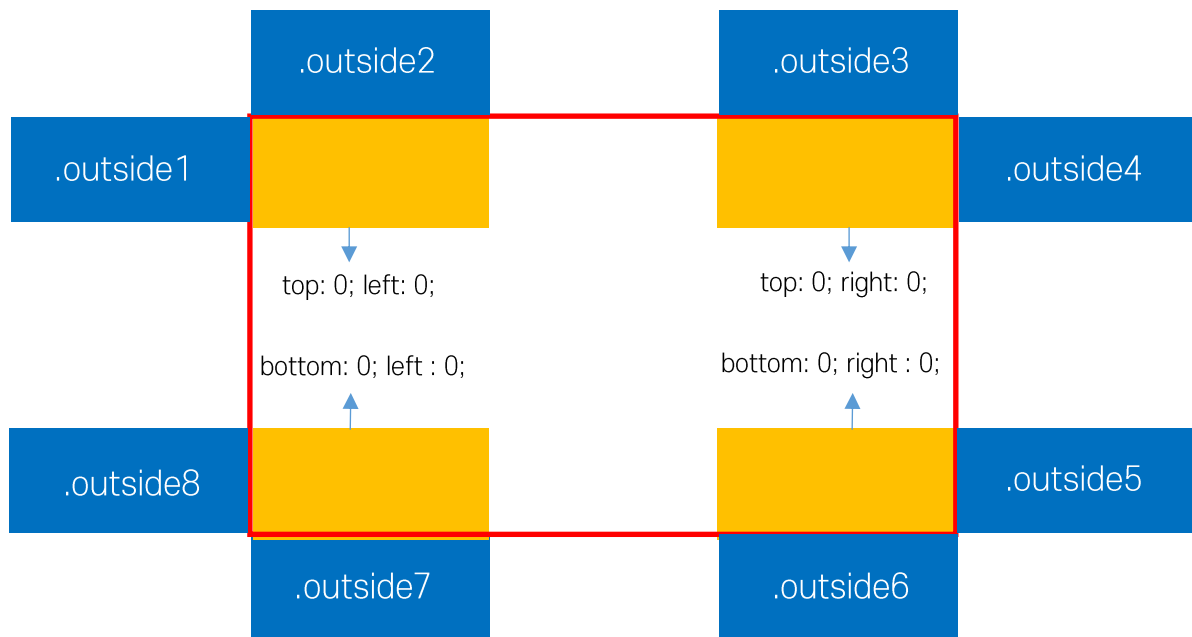
1. (z-index가 같다면) relative가 absolute 보다 우선한다.
2. (z-index가 같다면) HTML 순서상 나중에 나온 것이 우선한다.
3. absolute는 relative가 있는 부모요소에 종속되지만 fixed는 relative가 있는 부모요소와 관계 없이 브라우저를 부모요소로 위치함
4. top, bottom, right, left, z-index는 포지션 속성과 함께 사용

```
<div class="parent">  
  <div class="child-absolute"></div>  
  <div class="child-fixed"></div>  
</div>
```



## { Position 속성의 완벽한 이해와 활용 #02 }

자식요소를 부모요소 밖으로 코너나 좌우 상하 끝으로 위치시키기

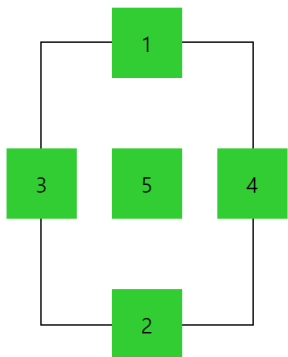


```
.outside1 {  
  top: 0;  
  left: -200px; /*right: 100%;*/  
}  
.outside2 {  
  top: -100px; /*bottom: 100%;*/  
  left: 0;  
}  
.outside3 {  
  top: -100px; /*bottom: 100%;*/  
  right: 0;  
}  
.outside4 {  
  top: 0;  
  right: -200px; /*left: 100%;*/  
}
```

```
.outside5 {  
  bottom: 0;  
  right: -200px; /*left: 100%;*/  
}  
.outside6 {  
  bottom: -100px; /*top: 100%;*/  
  right: 0;  
}  
.outside7 {  
  bottom: -100px; /*top: 100%;*/  
  left: 0;  
}  
.outside8 {  
  bottom: 0;  
  left: -200px; /*right: 100%;*/  
}
```

## { Position 속성의 완벽한 이해와 활용 #03 }

자식요소를 부모요소 수평 수직 중심에 유동적으로 위치시키기



```
<div class="parent">
  <div class="child">1</div>
  <div class="child">2</div>
  <div class="child">3</div>
  <div class="child">4</div>
  <div class="child">5</div>
</div>
```

HTML

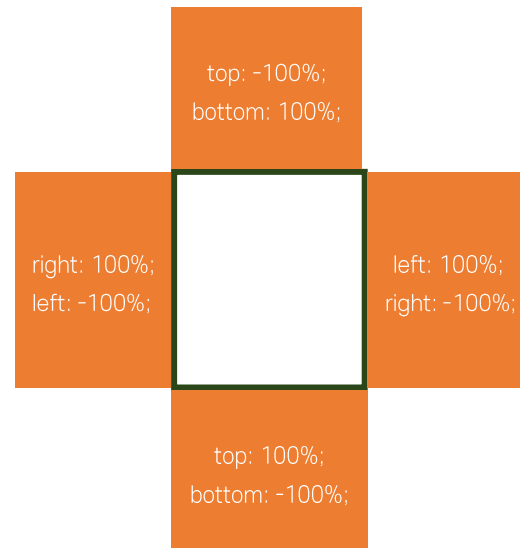
```
.parent {
  border: 2px solid;
  width: 300px; height: 400px;
  position: absolute;
  top: 50%; left: 50%;
  transform: translate(-50%, -50%);
}
.child {
  text-align: center;
  line-height: 100px; font-size: 30px;
  background-color: limegreen;
  width: 100px;
  height: 100px;
  position: absolute;
}
```

CSS

bottom: 0 대신 top: 100%를 사용할 경우  
transform: translate(-50%, -50%);  
right: 0 대신 left: 100%를 사용할 경우  
transform: translate(-50%, -50%);

```
.child:nth-child(1) {
  left: 50%; top: 0;
  transform: translate(-50%, -50%);
}
.child:nth-child(2) {
  left: 50%; bottom: 0;
  transform: translate(-50%, 50%);
}
.child:nth-child(3) {
  left: 0; top: 50%;
  transform: translate(-50%, -50%);
}
.child:nth-child(4) {
  right: 0; top: 50%;
  transform: translate(50%, -50%);
}
.child:nth-child(5) {
  top: 50%; left: 50%;
  transform: translate(-50%, -50%);
}
```

CSS



▲ 부모요소의 상하좌우 끝에 보내는 좌표 속성



## { Display 속성의 완벽한 이해와 활용 }

### CSS 퍼블리싱의 모든 것! HTML 인라인요소, 블록요소, 인라인블록

#### ■ 인라인(inline) 요소

- ✓ 한 줄에 여러 개 배치
- ✓ 기본 너비 값은 콘텐츠의 너비 값
- ✓ 크기 값을 가질 수 없음
- ✓ 상하 마진은 가질 수 없음

#### ■ 블록(block) 요소

- ✓ 한 줄에 한 개만 배치
- ✓ 기본 너비 값은 100%
- ✓ 크기 값을 가질 수 있음
- ✓ 상하좌우 마진 가질 수 있음

#### ■ 인라인 블록(inline-block) 요소

- ✓ 한 줄에 여러 개 배치
- ✓ 기본 너비 값은 콘텐츠의 너비 값
- ✓ 크기 값을 가질 수 있음
- ✓ 상하좌우 마진 가질 수 있음

■ 인라인(inline) 요소 - span, a, small, big, em, u, ins, i, s, del, br, q, b, strong, mark, sub, sup, label, map, cite, abbr, time

■ 블록(block) 요소 - div, p, table, blockquote, ul, ol, li, dl, td, dd, table, tr, th, form, hr, h1~h6, header, nav, footer, main, section, article, aside, figure, figcaption, summary, details, address

■ 인라인 블록(inline block) 요소 - img, input, button, textarea, select, 대부분의 Font Icon 태그, video, audio, iframe

- 블록 요소 중앙 정렬 : 중앙 정렬이 될 자신에 margin: auto (단, 블록요소만 margin: auto를 사용할 수 있음)
- 인라인 요소 및 인라인블록 요소 중앙 정렬 : 부모요소에 text-align 속성으로 정렬
- 모든 요소(인라인요소, 블록 요소, 인라인블록 요소) 왼쪽 배치 오른쪽 배치 : float 속성으로 배치
- position: absolute 또는 position: fixed를 주면 모든 요소는 인라인 블록 요소로 display 속성이 변경됨.

display 속성을 확인하기 위해 가장 좋은 방법은 요소를 2개 이상 생성하고 보더를 그려서 확인합니다.

→ 인라인 요소 : 주로 서식을 꾸미는 태그들

- :before :after는 인라인 요소

→ 블록 요소 : 주로 레이아웃을 꾸미는 태그들과 목록

→ 인라인블록 요소 : 주로 폼 요소 구성 태그들과 비디오, 오디오

- 5픽셀 내외의 우측 마진, 이미지 요소 아래 마진 있음

→ 부모요소에 flex가 선언되면 모든 엘리먼트는 display 속성 관계 없이 자식요소는 margin이 없는 인라인 블록으로 변경

▲ 코딩웍스 유튜브 영상 참고 : <https://youtu.be/qTgqyNRcos0>

# 🎯 [HTML+CSS+JS 포트폴리오 실전 퍼블리싱] HTML 구조 와이어프레임

## { 퍼블리싱의 시작이자 완성! HTML 구조 와이어프레임 만들기 #01 }

퍼블리싱 2~3개월 공부한 학생들도 HTML 구조가 주어진다면 세부적인 UI 퍼블리싱은 잘 하는 편입니다. 하지만 HTML 구조를 주지 않고 UI 디자인을 보고 만들라고 하면 무척 어려워 합니다. 그래서 'HTML 구조를 잘 만드는 것'이 퍼블리싱의 시작이자 완성입니다.

UI 디자인을 보고 와이어프레임을 만들 때 중요한 3가지입니다.

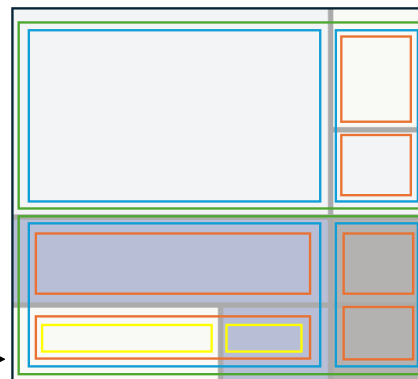
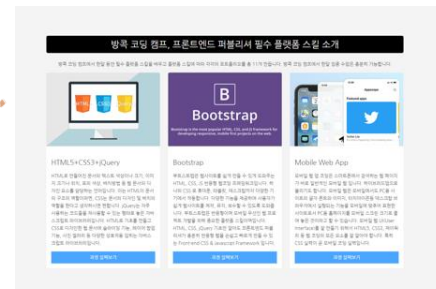
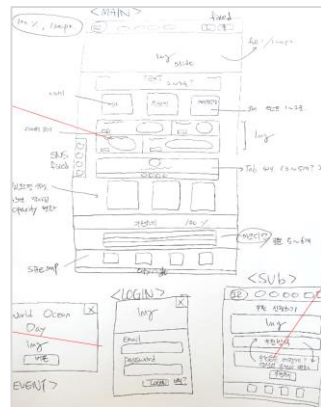
- 첫번째, 부모요소와 자식요소의 관계를 잘 구분해서 만듭니다.
- 두번째, 가로 배치되는 HTML 엘리먼트들을 부모요소로 반드시 감싸는 구조로 만듭니다.
- 세번째, HTML 구조를 만들면서 적절한 클래스 이름을 부여하면서 만듭니다.

① NotePad++로 만듭니다.

② 태그 상관하지 않고 무조건 div로 계층구조를 정확히 만듭니다.

③ 2)번 만든 후 적절한 태그와 클래스 이름을 부여해서 HTML 구조를 완성합니다.

④ 박스가 그려진 UI 디자인과 완성된 HTML 구조를 보고 코딩합니다.



```
.container
  .up
    .item-left
    .item-right
      div
      div
  .down
    .item-left
      div
    .item-sub
      div
      div
    .item-right
      div
      div
```



# [HTML+CSS+JS 포트폴리오 실전 퍼블리싱] HTML 구조 와이어프레임

## 필독사항



## [필독] HTML 와이어프레임 구조 보더 박스 및 클래스 네이밍 요령 #02

### HTML 와이어프레임(Wireframe)이란?

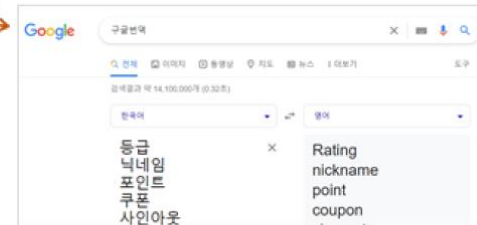
- 퍼블리싱에서 모든 것들 중에서 가장 중요하고 우선이 되어야 함.
- 텍스트에디터로 코딩하지 않고 HTML 구조 파악을 위한 뼈대를 말함
- HTML 와이어프레임 구조 만들 때 HTML 태그 뿐만 아니라 클래스네임도 함께 표시
- HTML 와이어프레임 구조 제대로 만드는 것이 퍼블리싱을 잘하기 위한 필수 작업
- HTML 와이어프레임 구조를 잘 만드려면 CSS 실력이 좋아야 함
- HTML 와이어프레임 구조를 만들 때 반드시 웹표준을 지켜가며 만들어야 합니다.  
ex) 인라인 요소가 블록 요소를 감싸면 안됨 (단, a태그는 제외)  
ex) 블록 요소 내에 블록 요소를 넣으면 안됨 (단, div 제외)
- 최대한 다양한 HTML 태그를 활용한다.
- 일관성 있는 HTML 계층 구조를 사용해야 한다.

### HTML 와이어프레임 보더 박스(Border Box) 그릴 때 요령

- 가로 배치의 경우, 2개 이상인 경우 반드시 부모 요소로 감싼다.
- 세로 배치의 경우, 내용 상 같은 구조를 부모 요소로 감싼다.
- 같은 색상은 같은 계층으로 형제요소(색상의 종류는 관계없음)
- 가장 겹쳐 있는 검색색 보더는 .container는 편의상 그리지 않고 .container 바로 아래 inner라는 클래스 네임이 있는 것부터 그림
- 처음부터 박스를 그리지 않고 HTML 와이어프레임을 잘 만드는 일은 어려운 일입니다.

### HTML 와이어프레임 구조 만들 때 클래스 네임 만들기

- 클래스 네임 만들기 곧, Class Naming은 경력자에게도 꽤 힘든 일
- 클래스네임은 제작자의 성향에 따라 다르기 때문에 정답은 없습니다. 하지만 클래스네임은 조금 길어지더라도 이름을 보고 미루어 짐작할 수 있도록 만들도록 노력해야 함
- 단어가 중복되는 경우 대시(-) 또는 언더바(\_)로 구분합니다. 개인적으로 대시 추천  
ex) 마이페이지에서 사용자 정보라면 user + info = .user-info  
ex) 주문하기 버튼(button)이라면 btn + order = button,btn-order  
ex) 음식종류 div 라면 food + type = .food-type
- 클래스 네임을 만들 때 구글 번역을 활용하면 합리적인 Class Naming이 가능



### [매우 중요 사항]

추후 영상을 통해서 코딩웍스와 HTML 와이어프레임 구조를 상세히 만들지만 수강생분들이 먼저 UI 디자인을 보고 혼자서 HTML 와이어프레임 구조를 만들어보세요. 처음에 어려울 것 입니다. 하지만 스스로의 이런 과정이 없으면 HTML 와이어프레임 구조 만들기 훈련이 되지 않습니다. 꼭! 혼자 힘으로 먼저 만들어 보시고 영상을 통해서 부족한 부분을 알아가는 시행착오의 과정을 거쳐야 합니다.

.inner : 요소의 안쪽을 묶을 때

ex) .section-inner

.outer : 요소의 바깥쪽을 묶을 때

ex) .section-outer

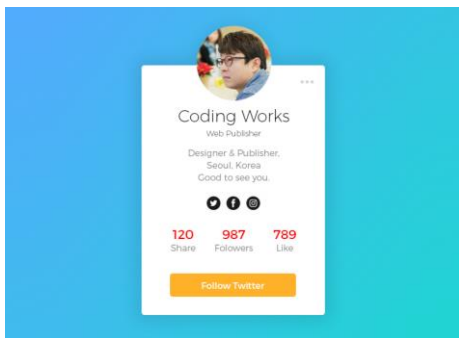
.items : 여러 개의 묶음으로된 자식요소를 묶을 때

ex) .card-items

.item : 개별적인 요소를 하나로 묶을 때

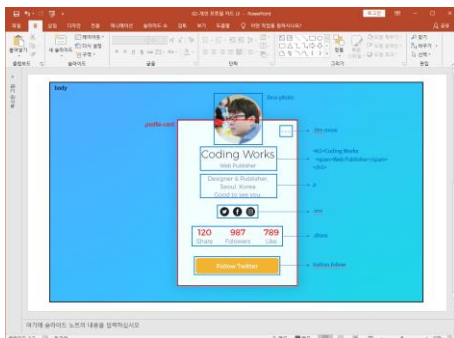
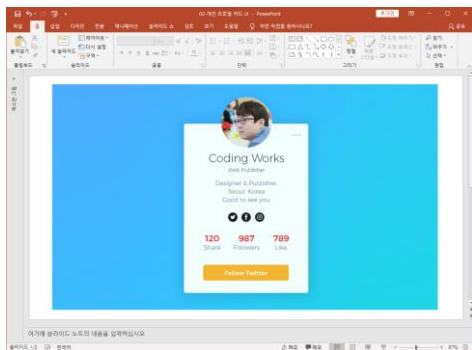
ex) .card-item

## { 퍼블리싱의 시작이자 완성! HTML 구조 와이어프레임 만들기 #03 }



▲ 제작할 UI 디자인 캡처를 준비하고 캡처 도구로 필요한 부분만 캡처합니다.

▼ ① 핸드 드로잉을 A4 용지에 러프하게 그리기 또는 PPT에 밀 그림 넣기



▲ ② 제작할 UI 캡처를 PPT에 넣고 색상이 다른 박스로 와이어 프레임 그리기

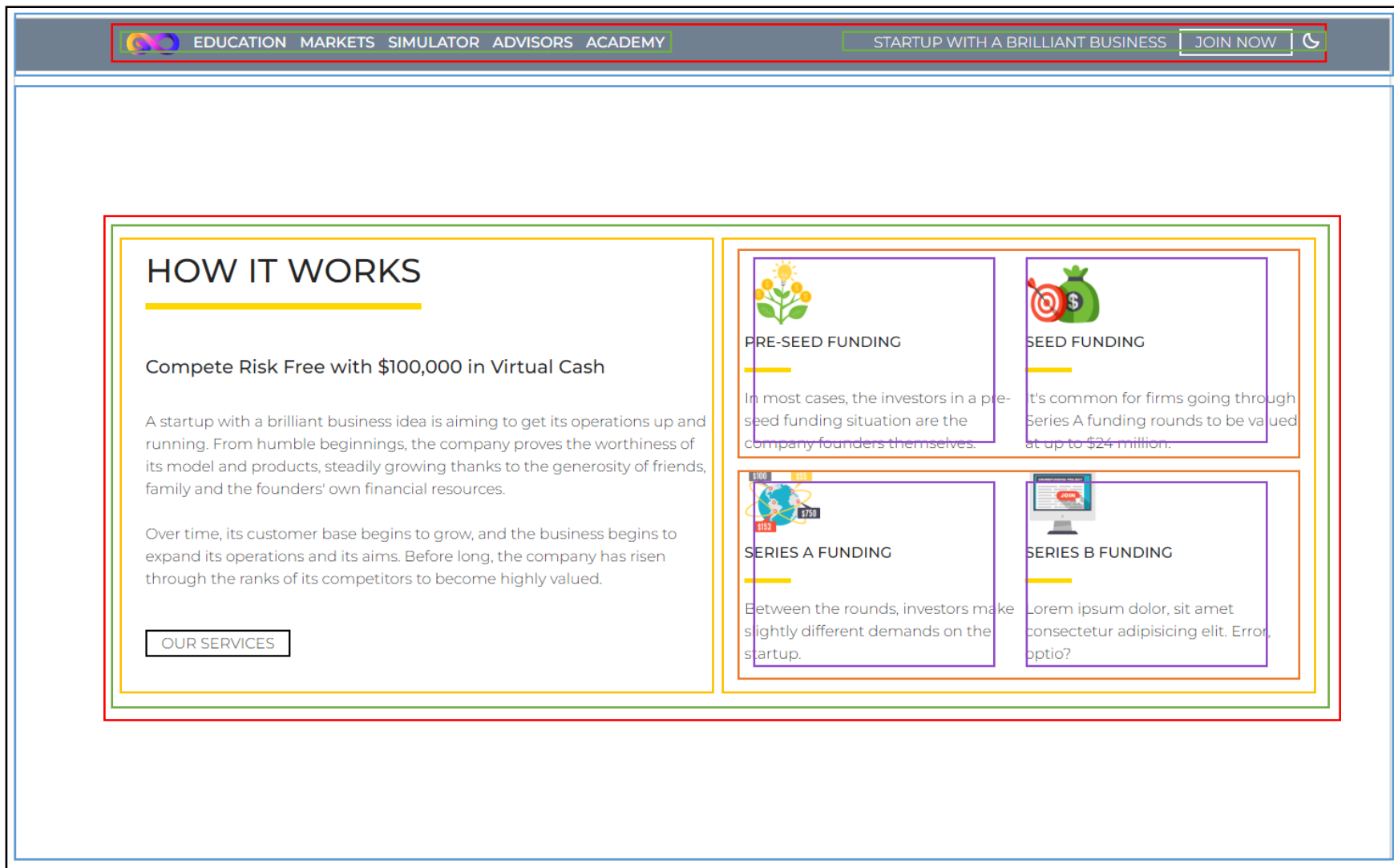
▼ ③ PPT 그림을 기준으로 NotePad++에서 클래스명과 함께 HTML 구조 만들기

```
.profile-card
  .face-photo
    img
  h1 > span
  p
  .sns
    a > font-icon
    a > font-icon
    a > font-icon
```

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>HTML 구조 와이어 프레임 미션(2) </title>
6 </head>
7 <body>
8
9   <div class="profile-card">
10     <div class="face-photo">
11       
12     </div>
13     <h1>Coding Works <span>Web Publisher</span></h1>
14     <p>
15       Designer & Publisher, Seoul, Korea <br>
16       Good to see you.
17     </p>
18     <div class="sns">
19       <a href="#"><i class="fa fa-twitter"></i></a>
20       <a href="#"><i class="fa fa-facebook"></i></a>
21       <a href="#"><i class="fa fa-instagram"></i></a>
22     </div>
23   </div>
24
25 </body>
26 </html>
```

▲ ④ PPT와이어 프레임 기준으로 텍스트 에디터에서HTML 구조 코딩하기

## #01. 다크모드(Dark Mode) 웹페이지



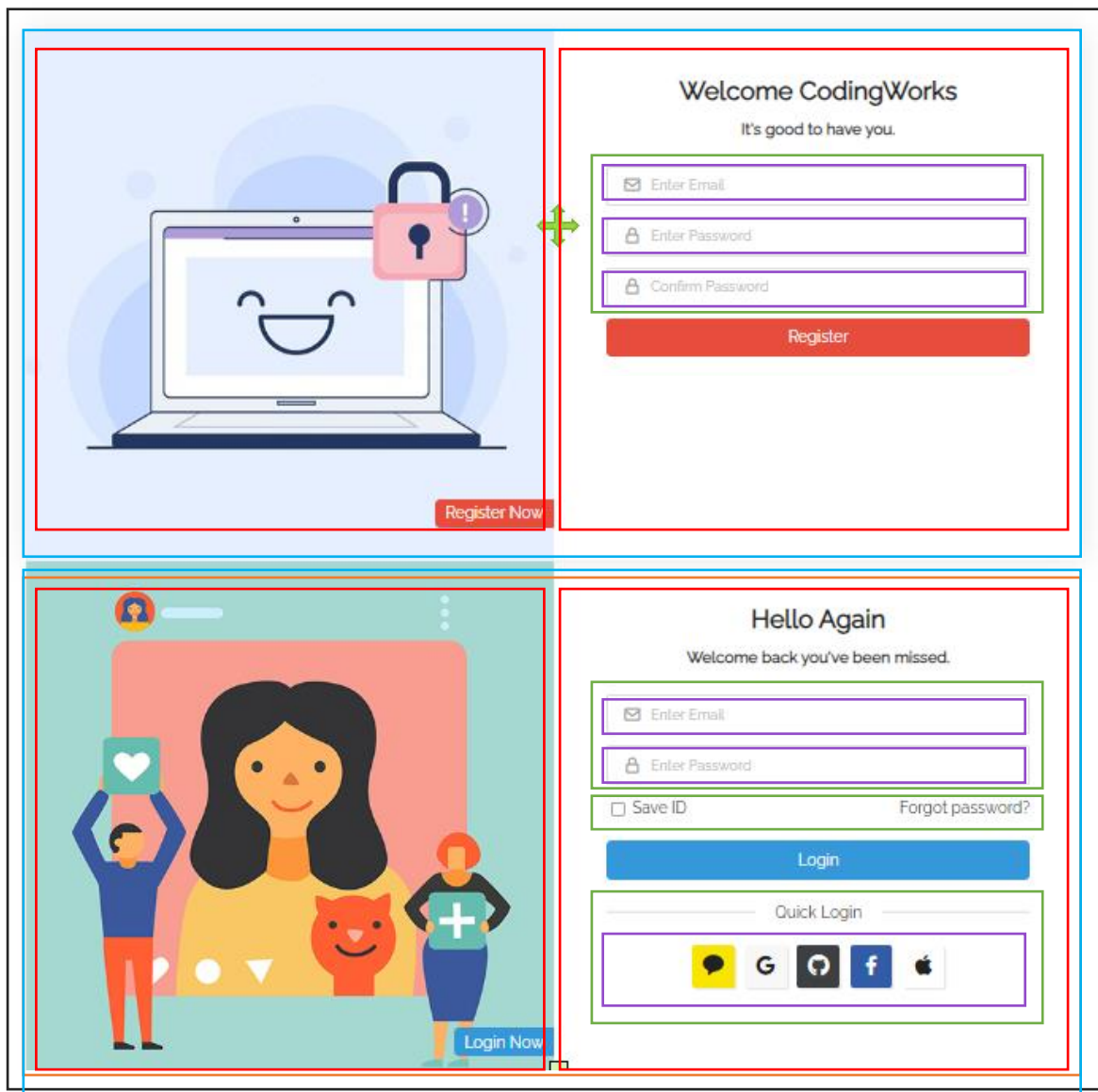
```
.container
  header
    nav
      .logo
        a > img
      .gnb
        a * 5
      .lnb
        a.btn-more
        .mode
  section
    .section-inner
      .about-items
        .info
          h1.headline
          h4
          p * 2
          a.btn-more
        .skill
          .skill-items
            .skill-item
              img
              h1.headline
              p
            .skill-item
              img
              h1.headline
              p
            .skill-items
              .skill-item
                img
                h1.headline
                p
              .skill-item
                img
                h1.headline
                p
```



# 🎯 [HTML+CSS+JS 포트폴리오 실전 퍼블리싱] HTML 구조 와이어프레임 실전 제작

코딩웍스(Coding Works)

## #02. 트랜지션 교차 로그인 회원가입



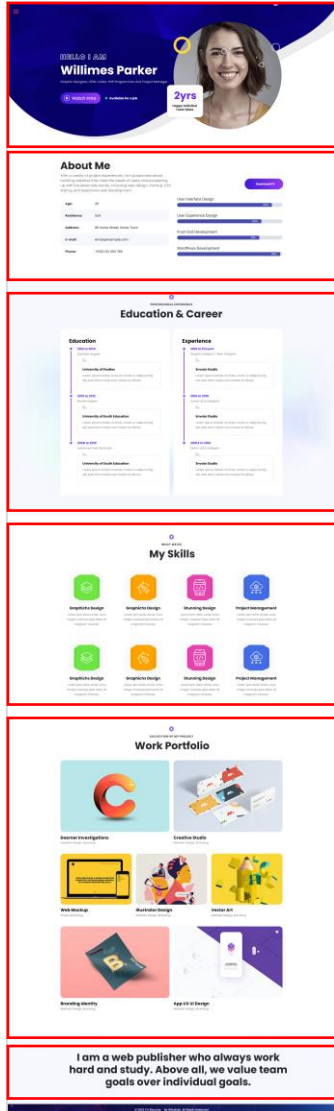
```
.swap-login-form
.swap-content.register
  .photo
    img
    button.btn-swap
  .swap-form
    h1 > span
    .fields
      .field
        input[type=email]
        font-icon
      .field
        input[type=password]
        font-icon
      .field
        input[type=password]
        font-icon
    button
.swap-content.login
  .photo
    img
    button.btn-swap
  .swap-form
    h1 > span
    .fields
      .field
        input[type=email]
        font-icon
      .field
        input[type=password]
        font-icon
    .status
      label
        input[type=checkbox]
      a
    button
  .quick-login
    span.or
    .sns-items
      (a > font-icon) * 5
```

# [HTML+CSS+JS 포트폴리오 실전 퍼블리싱] HTML 구조 와이어프레임 실전 제작

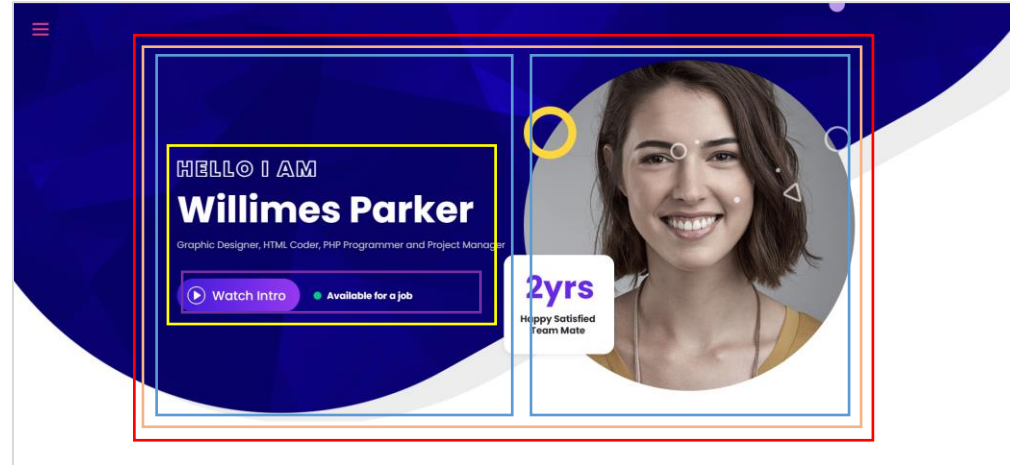
코딩웍스(Coding Works)

#03. 개인 이력서 포트폴리오 홈페이지

1



```
.container
  section.intro
  section.about
  section.edu-career
  section.skills
  section.work-portfolio
  section.epilogue
  footer
```



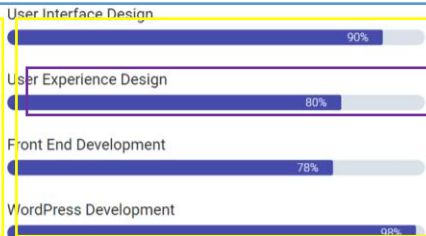
```
section.intro
  .section-inner
    .section-content
      .name
        div
          h1 > span
          p
          .watch-intro
            button.btn-watch
            span
      .photo
        img
        .yrs
        h2
        span
```

## About Me

With a variety of project experiences, I am passionate about building websites that meet the needs of users, always keeping up with the latest web trends, including web design, markup, CSS styling, and responsive web development.

Download CV

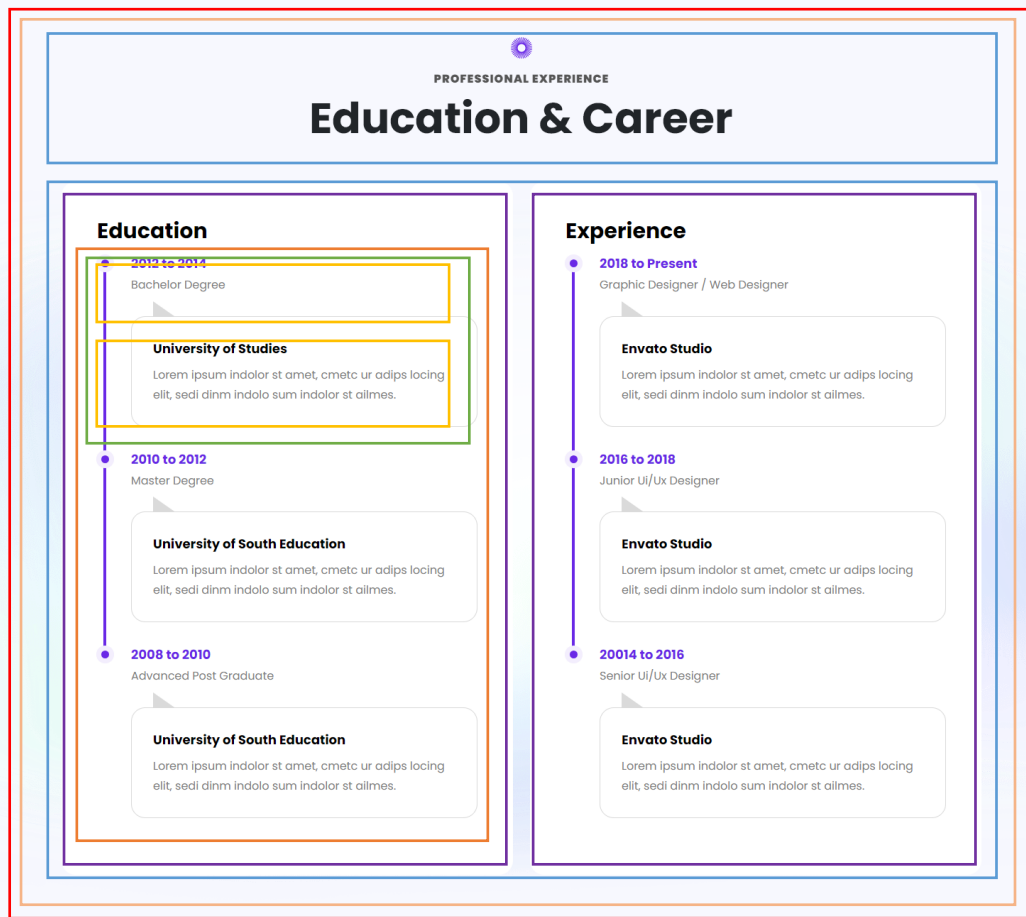
**Age:** 29  
**Residence:** USA  
**Address:** 88 Some Street, Some Town  
**E-mail:** email@example.com  
**Phone:** +0123 123 456 789



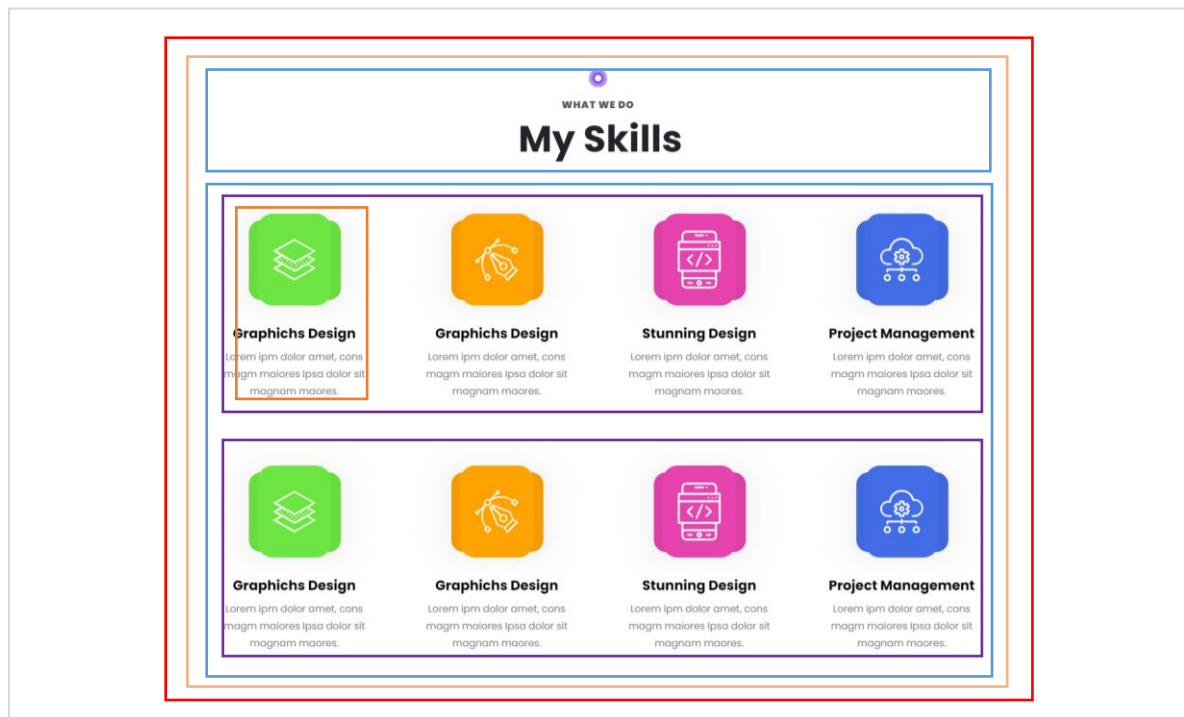
```
section.about
  .section-inner
    .section-content
      .profile
        div
          h1.headline
          p
        div
          button.download-cv
      .personal-ability
        ul.personal
          li > b
          li > b
          li > b
          li > b
          li > b
        .ability
          .ability-item
            em
            div
              span
          .ability-item
            em
            div
              span
          .ability-item
            em
            div
              span
          .ability-item
            em
            div
              span
```







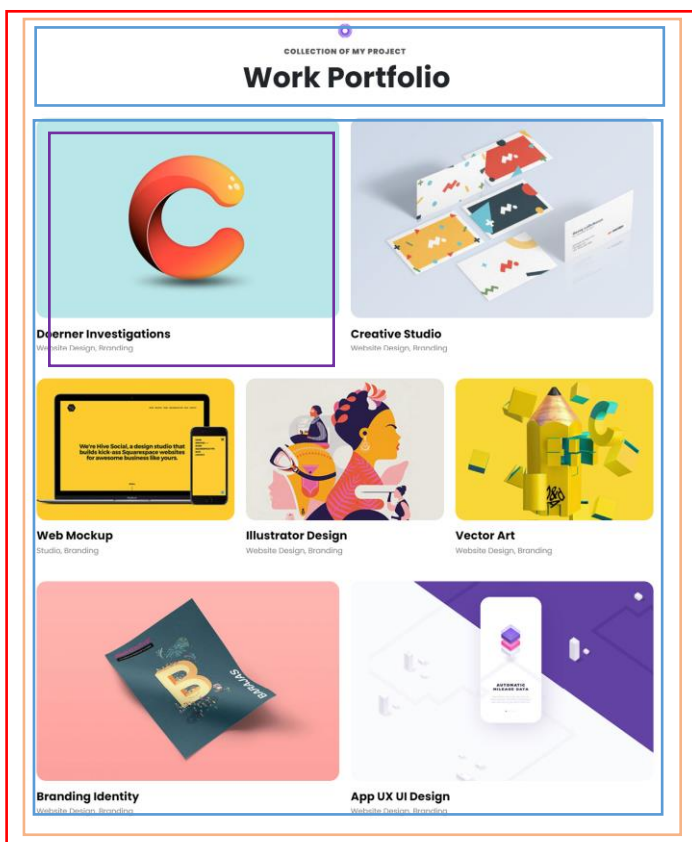
```
section.edu-career
  .section-inner
    .section-content
      .section-index
        span
          h1.headline
      .timeline
        .edu-timeline
          h3
            .timeline-items
              .timeline-item
                .title
                  b
                span
                  .detail
                    b
                    p
        .career-timeline
          h3
            .timeline-items
              .timeline-item
                .title
                  b
                span
                  .detail
                    b
                    p
```



```
section.skills
  .section-inner
    .section-content
      .section-index
        span
        h1.headline
      .skills-items-outer
        .skills-items
          .skills-item
            img
            h4
            p
        .skills-items
          .skills-item
            img
            h4
            p
```

## #03. 개인 이력서 포트폴리오 홈페이지

5



```
section.portfolio
  .section-inner
    .section-content
      .section-index
        span
        h1.headline
      .portfolio-items
        .portfolio-item
          .thum
            img
          .caption
            h4
            p

section.epilogue
  .section-inner
    .section-content
      h1.focus

footer
  p
```