



3주차 CSS

☼ 상태	승인
≡ 설명	CSS 선택자 , 단위 ,reset.css

CSS 를 활용하면 뼈대만 있는 HTML 에 스타일을 입히는 것이 가능하다.

CSS 의 기본 문법은 아래와 같다.

```
선택자 {  
  속성 : 속성값;  
  속성 : 속성값;  
}
```

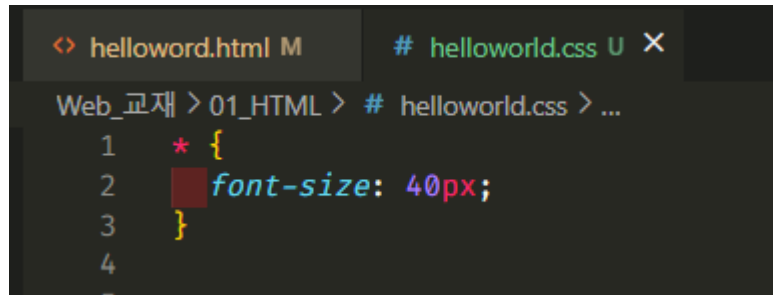
class 혹은 id 혹은 태그 자체의 이름으로 해당 요소를 선택 하고 { } 안에 입히고 싶은 스타일을 넣으면 된다.

속성을 설정할때에는 반드시 ; 를 붙여서 마무리 해줘야 한다.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <title>Product Form</title>  
  </head>  
  <body>  
    <div class="title">제목</div>  
    <p class="content">  
      Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam  
      totam debitis nesciunt eaque alias facere molestias id rerum a quidem
```

```
laboriosam natus ipsa officiis et magni vel laborum!  
</p>  
</body>  
</html>
```

다음과 같은 `html` 을 생성해보자.



이후 `helloworld.css` 파일을 만든 후, 내용을 위처럼 따라 적어보자. `font-size`는 글씨의 크기를 바꿀 수 있는 속성으로 자세한 내용은 나중에 알아보자.

해당 `CSS` 파일을 `HTML` 에 연동하기 위해서는 `HTML` 파일의 `<head>` 태그 안에 `<link>` 태그를 활용하여 연결해줄 수 있다.

```
<head>  
  <meta charset="UTF-8" />  
  <title>Product Form</title>  
  <link rel="stylesheet" href="helloworld.css" />  
</head>
```

아래와 같이 속성이 잘 적용되는걸 알 수 있다.

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!

선택자

원하는 요소를 선택하는 부분이다.

전체 선택자

```
*{  
    font-size : 40px;  
}
```

모든 요소를 선택하는 선택자이다.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <title>Product Form</title>  
  </head>  
  <body>  
    <div class="title blue">제목</div>  
    <p class="content" id="content">  
      Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam  
      totam debitis nesciunt eaque alias facere molestias id rerum a quidem  
      laboriosam natus ipsa officiis et magni vel laborum!  
    </p>  
  </body>  
  <style>  
    * {  
      font-size : 40px;  
    }  
  </style>  
</html>
```

이하 선택자 설명에서는 **style** 태그 안의 내용만 바뀌가면서 실습해 보자.

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!



<STYLE/> ?

HTML에서 바로 CSS를 사용할 수 있게 해주는 태그로, 태그 안에 CSS 문법에 맞게 작성하면 해당 부분이 바로 적용이 된다.

태그 선택자

태그를 기준으로 해당 태그들의 스타일을 변경할 수 있다.

```
<style>
  p {
    font-size : 40px;
  }
</style>
```

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!

클래스 선택자

클래스를 활용하여 요소들을 선택하고 변경할 수 있다.

. 을 앞에 붙임으로 뒤에 나오는 것이 `class` 임을 명시한다.

```
<style>
  .title {
    font-size : 40px;
  }
</style>
```

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!

아이디 선택자

아이디를 활용하여 요소를 선택하고 변경할 수 있다. 아이디는 각 요소당 하나씩만 가질 수 있으므로 1개의 요소만 선택하게 되는것이 특징

#을 앞에 붙임으로 뒤에 나오는 것이 id임을 명시한다.

```
<style>
  #content {
    font-size : 40px;
  }
</style>
```

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!

복합 선택자

위에서 사용했던 선택자들을 조합하거나 여러개 사용해서 요소를 선택할 수 있다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Product Form</title>
  </head>
```

```

<body>
  <div class="news">
    <div class="title blue">제목</div>
    <p class="content" id="content">
      Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor
      ipsam totam debitis nesciunt eaque alias facere molestias id rerum a
      quidem laboriosam natus ipsa officiis et magni vel laborum!
    </p>
    <ul id="fruit-list">
      <li>사과</li>
      <li>배</li>
      <li>포도</li>
      <li>복숭아</li>
    </ul>
  </div>
  <p class="title">두번째 제목</p>
</body>
<style>
  .title {
    font-size: 40px;
  }
</style>
</html>

```

위와 같이 태그가 중첩되어 있는 경우를 가지고 복합 선택자가 무엇인지 알아보자.

태그선택자 + 클래스 선택자

두가지 선택자를 조합하여 아래와 같이 사용할 수 있다.

```

<style>
  p.title {
    font-size: 40px;
  }
</style>

```

제목

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor ipsam totam debitis nesciunt eaque alias facere molestias id rerum a quidem laboriosam natus ipsa officiis et magni vel laborum!

- 사과
- 배
- 포도
- 복숭아

두번째 제목

제목, 두번째 제목 모두 `title` 이란 `class` 명을 가지고 있지만 두번째 제목 부분만 `css` 가 적용된 것을 확인할 수 있다.

후손 선택자

아래와 같이 선택자들을 `space(공백)` 을 기준으로 구분하면 부모태그 아래의 태그들만 선택된다.

```
<style>
  .news .title {
    font-size: 40px;
  }
</style>
```

자식 선택자

```
<style>
  .news > .title {
    font-size: 40px;
  }
```

```
}  
</style>
```

선택자의 자식 중 하나를 선택할 수 있다.



후손 선택자 vs 자식 선택자

후손 선택자는 여러개의 태그가 중첩되어 있을때 자신의 후손들 중 해당 선택자에 맞는 태그가 있다면 CSS가 적용되지만 자식선택자는 자신의 직계 후손 태그에만 CSS가 적용된다.

(color 속성은 글씨의 색상을 정해주는 속성이다.)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Product Form</title>
  </head>
  <body>
    <div class="news">
      <div class="title blue">
        제목
        <div class="red">빨강으로 강조하고 싶은 부분</div>
      </div>
      <p class="content" id="content">
        Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor
        ipsam totam debitis nesciunt eaque alias facere molestias id rerum a
        quidem laboriosam natus ipsa officiis et magni vel laborum!
      </p>
      <ul id="fruit-list">
        <li>사과</li>
        <li>배</li>
        <li>포도</li>
        <li>복숭아</li>
      </ul>
    </div>
    <p class="title">두번째 제목</p>
  </body>
</html>
```

```
.news > .red {
  color: red;
}
```

```
.news .red {
    color: red;
}
```

위 두 가지 경우를 <STYLE> 태그 안에 넣고 어떻게 동작하는지 실습 하여 보자.



, 를 사용하면 하나의 **css** 를 두가지 선택자에 적용하는 것도 가능하다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Product Form</title>
  </head>
  <body>
    <div class="news">
      <div class="title blue">
        제목
        <div class="red">빨강으로 강조하고 싶은 부분</div>
      </div>
      <p class="content" id="content">
        Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ab, dolor
        ipsam totam debitis nesciunt eaque alias facere molestias id rerum a
        quidem laboriosam natus ipsa officiis et magni vel laborum!
      </p>
      <ul id="fruit-list">
        <li>사과</li>
        <li>배</li>
        <li>포도</li>
        <li>복숭아</li>
      </ul>
    </div>
    <p class="title">두번째 제목</p>
  </body>
  <style>
    .red,
    li {
      color: blue;
    }
  </style>
</html>
```

```
</style>
</html>
```

가상 클래스 선택자

hover

마우스가 요소 위에 올려져 있는 동안 실행되는 CSS를 지정

```
<style>
  .content:hover {
    font-size: 40px;
  }
</style>
```

마우스를 해당 요소에 올리는 경우에만 **CSS**가 적용되는 것을 확인할 수 있다.

Active

마우스가 클릭 되는동안 클릭한 요소에 CSS를 지정

```
<style>
  .content {
    color: red;
  }
  .content:active {
    font-size: 40px;
  }
</style>
```

CHILD

각 요소가 중복되어 계속 있는 경우 순서에 따라 **CSS** 를 설정할 수 있음



CHILD를 사용하는 이유 ?

ID, class를 사용하여 일일이 선택자를 연결해 주는 방법도 있지만, 프로그래머들에게 변수이름이나 클래스명을 짓는건 상당히 번거롭고 힘든 일이다.

```
<ul id="fruit-list">
  <li>사과</li>
  <li>배</li>
  <li>포도</li>
  <li>복숭아</li>
</ul>
```

위와 li 태그의 순서에 따라서 **CSS** 를 넣고 싶은 경우 **CHILD** 를 활용하면 보다 효율적으로 **CSS** 를 적용시킬 수 있다.

first-child

해당 요소의 첫번째 요소의 CSS를 설정

```
<style>
  #fruit-list li:first-child {
    color: red;
  }
</style>
```

NTH-CHILD

n번째 요소의 CSS를 설정할 수 있다.

```
<style>
  #fruit-list li:nth-child(3) {
    color: red;
  }
</style>
```

아래와 같이 사용하여 짝수 요소에만 CSS를 넣는 것도 가능하다.

```
<style>
  #fruit-list li:nth-child(2n) {
    color: red;
  }
</style>
```

이러한 선택자를 활용하면 넣고 싶은 부분에 CSS를 넣는 것이 가능하다. 또한 어떠한 선택자를 사용할 것인지, Id를 사용할 것인지 class를 사용할 것인지 정하는 것 또한 프로그래머의 역량이다.



추가 학습 ?

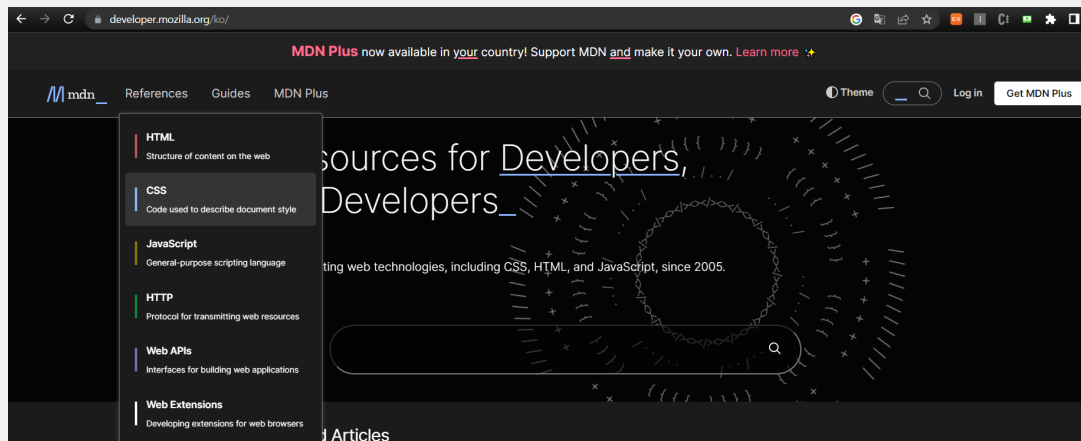
위에서 설명한 선택자 외에도 **인접형제 선택자** **일반형제 선택자** **NTH-OF-TYPE** **부정 선택자** **BEFORE** **AFTER** **속성 선택자** 등 여러가지 선택자가 존재한다. 다만 본 교재에서는 설명을 생략하도록 한다.

만약 위에서 설명했던 선택자에 대해 더 알아보고 싶거나, 설명되어 있지 않은 선택자에 알고싶다면

https://developer.mozilla.org/ko/docs/Learn/CSS/Building_blocks/Selectors#선택자의_유형

위사이트 혹은

구글에 MDN을 검색한 후, 사이트에 들어간 후



References - CSS 파트에 들어가서 필요한 항목을 검색하여 자세한 내용을 볼 수 있다.

우선순위

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Product Form</title>
  </head>
  <body>
    <div id="color_yellow" class="color_green" style="color: orange">
      Hello world
    </div>
  </body>
  <style>
    div {
      color: red;
    } /*태그선택자*/
    #color_yellow {
      color: yellow;
    } /*아이디선택자*/
    .color_green {
      color: green;
    } /*클래스 선택자*/
    div {
      color: blue;
    } /*태그 선택자*/
    * {
      color: darkblue;
    } /*전체선택자*/
    body {
      color: violet;
    } /* 상속*/
  </style>
</html>
```

만약 위와 같이 같은 요소에 css를 중복되게 설정하면 어떻게 될까? 해당 경우 우선순위에 따라서 CSS 적용이 결정되게 된다.

CSS 우선순위 규칙은 아래와 같다.

1. 명시도 점수가 높은 선언
2. 점수가 같은 경우, 가장 마지막에 해석되는 선언
3. 명시도는 '상속' 규칙보다 우선
4. !important가 적용된 선언 방식이 우선

이를 조금더 명확하게 점수로 표현할 수 있다. 해당 점수를 계산하는 방법은 아래와 같다.

1. !important
div {
 color : red !important;
} /* 가장 우선 */
2. 인라인 선언방식 (점수 1000)
<div style = "color : orange;">HELLO WORLD</div>
>>점수가 엄청크기도 해서 인라인 선언방식을 선호하진 않는다.
>>인라인 선언방식이란 HTML에서 바로 css를 선언해주는 방식이다.
3. 아이디 선택자 (점수 100)
#color yellow{
 color : yellow;
}
4. 클래스 선택자 (점수 10)
5. 태그 선택자 (점수 1)
6. 전체 선택자 (점수 0)

만약 후손 선택자를 활용한 상태여서 여러 선택자가 합쳐져 있다면 각 우선순위의 점수를 더 해주면 된다.

예를 들어 아래와 같은 CSS가 있다고 생각해보자.

```
div.fruit {
    color : blue;
}
```

위 태그의 우선순위 점수는 div(태그 ⇒ 1) + .fruit(클래스 ⇒ 10) = 11pt의 우선순위를 가지게 된다.



문제 !

아래 태그들의 점수는 몇점일까?

```
.list li.item { color : red; }
```

```
.list li:hover { color : red;}
```

```
#submit span { color : red;}
```

```
header .menu li:nth-child(2) { color : red; }
```



문제 1 정답 (꼭 풀어보고 확인하자)

```
.list li.item { color : red; } /* 21pt*/
```

```
.list li:hover { color : red;} /* 21pt*/
```

```
#submit span { color : red;}/*101pt*/
```

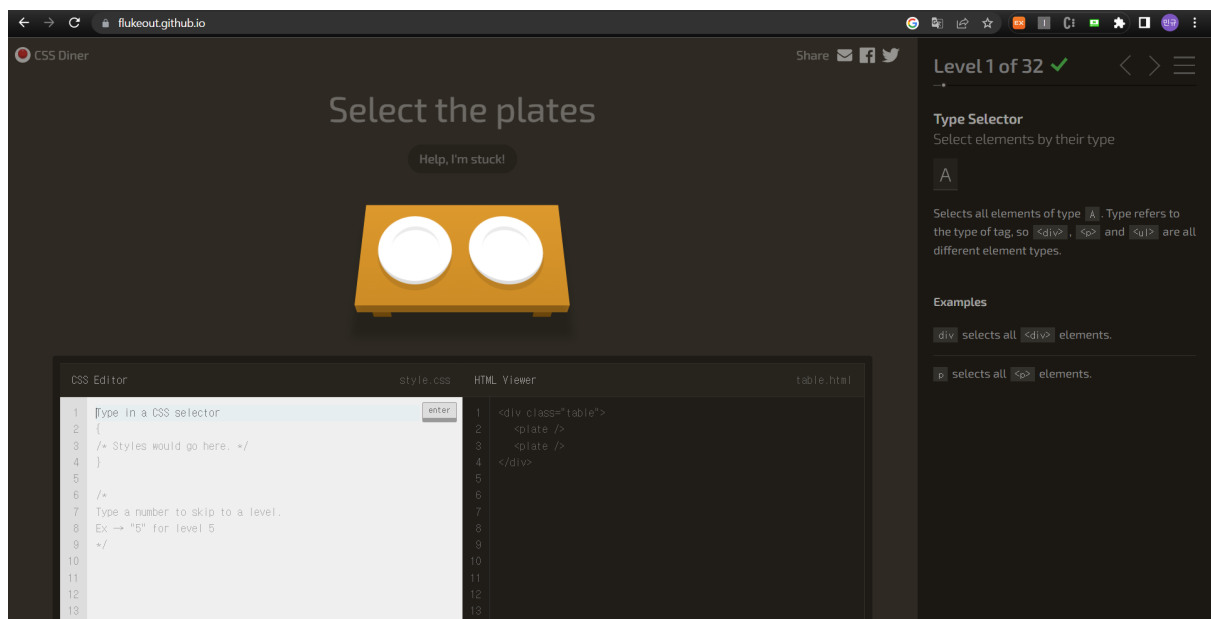
```
header .menu li:nth-child(2) { color : red; } /*22pt*/
```

선택자 연습하기

<https://flukeout.github.io/>

(CSS Diner 검색)

위 사이트를 이용하면 여러가지 선택자를 활용하여 CSS를 입힐 요소를 선택하는 것을 게임처럼 익혀볼 수 있다.



게임의 목표는 HTML 요소들 중에서 움직이는 요소들을 모두 한번에 담으면 되는 게임이다.

첫 번째 문제의 경우 HTML이 아래와 같이 주어져 있다.

```
<div class="table">
  <plate />
  <plate />
</div>
```

따라서 배웠던 태그 선택자를 활용하여

```
plate {  
  
}
```

끝이라면 접시 부분을 모두 선택할 수 있을 것이다 따라서 답안을 적는 부분에 plate를 입력하면 된다.

페이지에 이동해서 11단계까지 해결해보자.

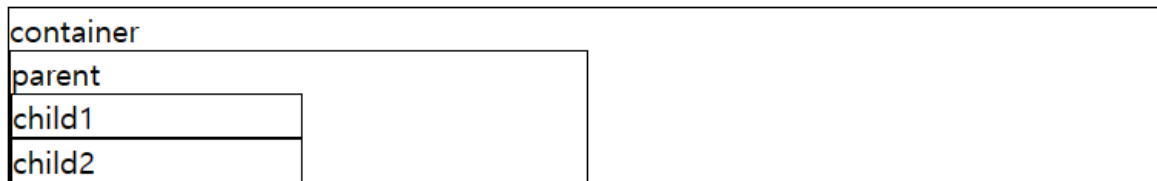
단위

css에서 사용되는 단위에 대해 알아보자.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <title>Product Form</title>  
  </head>  
  <body>  
    <div class="container">  
      container  
      <div class="parent">  
        parent  
        <div class="child">child1</div>  
        <div class="child">child2</div>  
      </div>  
    </div>  
  </body>  
  <style>  
    /* 스타일 작성 */  
  </style>  
</html>
```

px

```
body * {  
  border: 1px solid black;  
}  
.container {  
  width: 600px;  
}  
.parent {  
  width: 300px;  
}  
.child {  
  width: 150px;  
}
```



px단위로 width를 설정해 줄 수 있다.



border , width ?

border은 요소의 테두리를 정해주는 속성, width는 가로 길이를 정해주는 속성이다. (자세한건 속성 파트에서 다루도록 하자)

%

```
<style>  
  body * {  
    border: 1px solid black;  
  }
```

```

.container {
  width: 600px;
}
.parent {
  width: 50%;
}
.child {
  width: 30%;
}
</style>

```

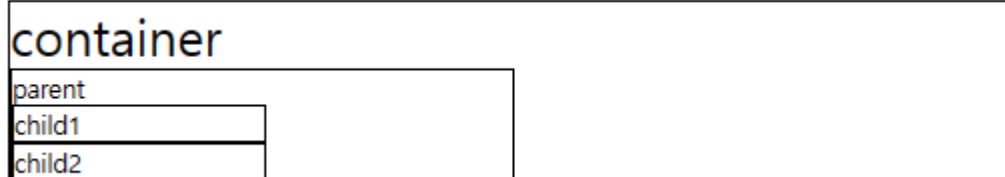


em

```

<style>
  body * {
    border: 1px solid black;
  }
  .container {
    width: 20em;
    font-size: 20px;
  }
  .parent {
    width: 20em;
    font-size: 10px;
  }
  .child {
    width: 10em;
  }
</style>

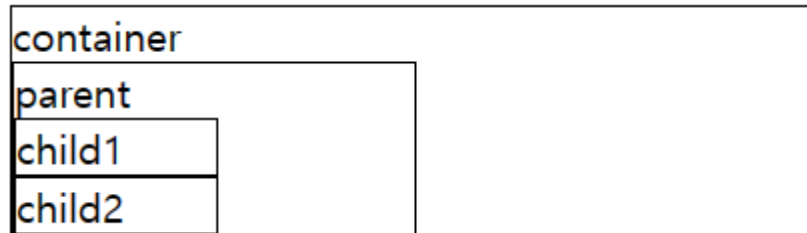
```



rem

rem의 경우 컴퓨터 기준으로 1rem === 16px인 경우가 대다수이다. 하지만 핸드폰이나 다른 특정기기에서는 14px이나 10px로 지정되어 있는 경우가 존재하고, 따라서 사용하는 기기에 따라 보여지는 글꼴의 크기를 다르게 설정할 수 있다.

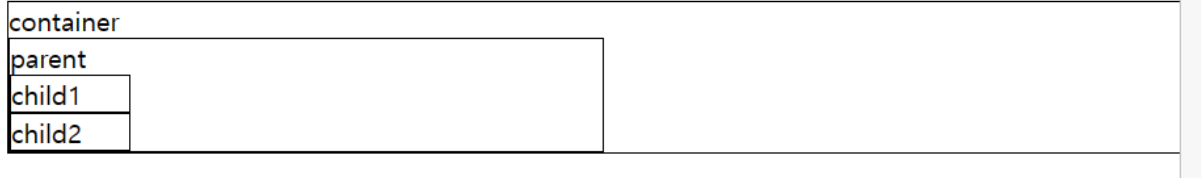
```
<style>
  body * {
    border: 1px solid black;
  }
  .container {
    width: 20rem;
  }
  .parent {
    width: 10rem;
  }
  .child {
    width: 5rem;
  }
</style>
```



vw, vh

```
<style>
  body * {
    border: 1px solid black;
  }
  .container {
    width: 100vw;
  }
  .parent {
    width: 50vw;
  }
  .child {
    width: 10vw;
  }
```

```
}  
</style>
```



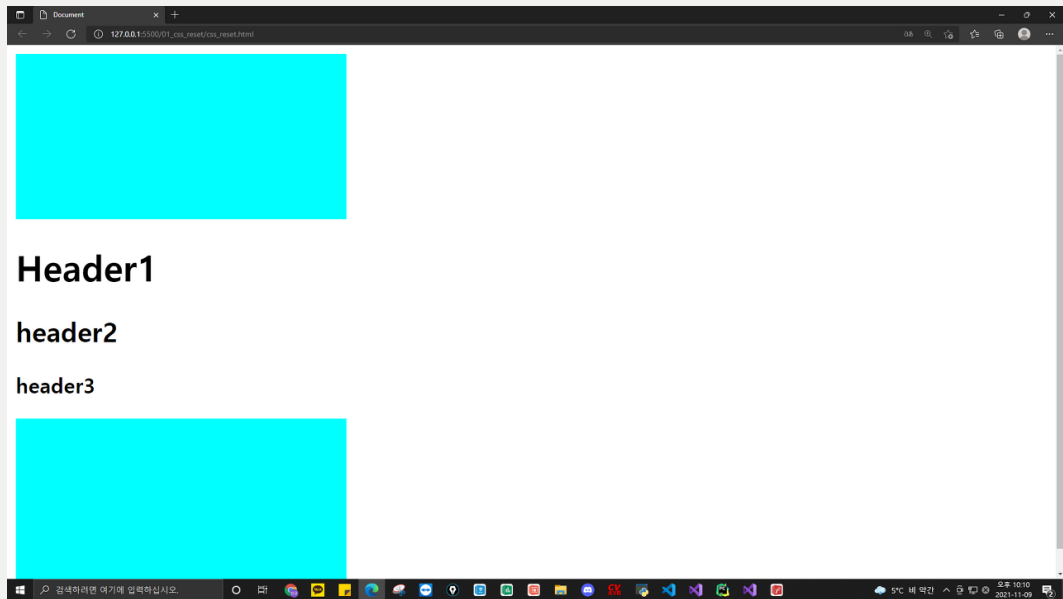
내용을 정리하면 아래와 같다.

- 1 px : 화면의 가장 작은 단위인 픽셀을 기준으로 css를 설정
- 2 % : 본인의 부모 요소의 값을 100%를 기준으로 하여 css를 설정
- 3 em : font 크기에 따른(font-size) 단위
- 4 rem : 루트 글꼴 크기에 따른 단위 (기본값 16px)
- 5 vw, vh : viewpoint(웹브라우저의 크기)의 값



RESET.CSS

웹브라우저(크롬,웨일,엣지,사파리 등) 마다 태그에 따른 고유한 CSS를 가지고 있다.



p 태그나 **Header** 태그들을 사용하면 위처럼 기본적인 여백을 가지게 된다. 이렇게 정해져 있는 CSS는 때로는 개발자들의 목적과 다르게 적용될 때가 있다.

이를 reset.css라고 하며 적용시키는 방법은 다양하지만 그중 한가지를 실습해보자.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Product Form</title>
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"
    />
    <!--순서가 먼저되어야 한다.-->
  </head>
  <body>
```



```
<header>헤더헤더</header>  
</body>  
</html>
```

위와같이 `link` 태그를 활용하여 `reset.css` 를 적용하면 헤더 영역에 기본적으로 가지고 있던 `css` 는 적용되지 않게 된다.