

2강

CSS 기초 교육

Index.html

HTML

- 페이지의 뼈대, 정보를 구성

+

Style.css

CSS

- 페이지 장식 및 꾸밈

CSS

Cascading Style Sheets

| CSS를 HTML에 적용하기

1. 인라인 스타일 (Inline style)
2. 내부 스타일 시트 (Internal style sheet)
3. 외부 스타일 시트 (External style sheet)

1. 인라인 스타일 (Inline style)

html 태그 내부에 스타일을 적용하는 방법

```
<body>
  <!-- 다이어리 타이틀 -->
  <header>
    <h1 style="color: red">나의 다이어리</h1>-->
  <nav>
    <p>
      홈<br/>
```



| 2. 내부 스타일 시트 (Internal style sheet)

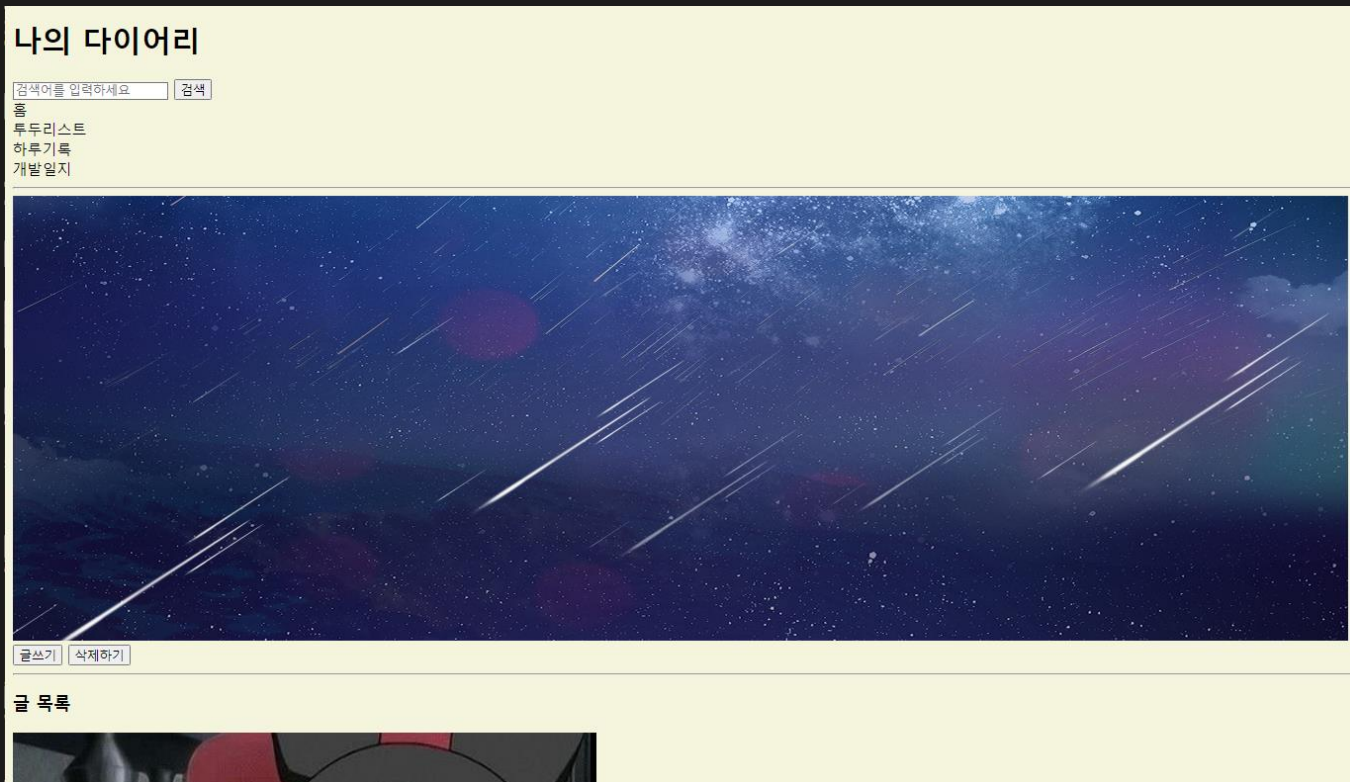
html 문서 안에 <style>태그를 사용하여 적용하는 방법

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>나의 다이어리</title>

  <style>
    body {
      background-color: beige;
    }
  </style>
</head>
```

| 2. 내부 스타일 시트 (Internal style sheet)

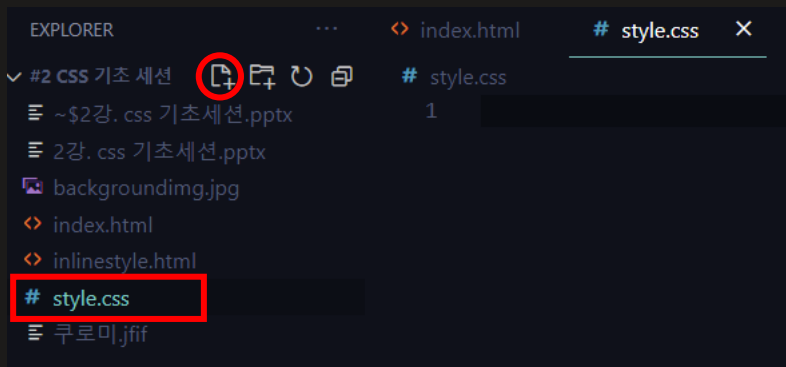
html 문서 안에 <style>태그를 사용하여 적용하는 방법



| 3. 외부 스타일 시트 (External style sheet)

html 문서 외부에 css 파일을 생성하여 적용하는 방법

1. css 파일 생성하기



2. html 파일과 연결하기

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>나의 다이어리</title>
  <link rel="stylesheet" href="./style.css">
</head>
```

`<link rel="stylesheet" href="css파일주소">`

| 3. 외부 스타일 시트 (External style sheet)

html 문서 외부에 css 파일을 생성하여 적용하는 방법

3. css 파일 사용하기

```
선택자 {  
  
    프로퍼티 : 값;  
  
}
```

CSS 구성 - 스타일 선택자

전체 선택자 : *	<pre>* { color: blue; }</pre>
태그 선택자 : 태그이름	<pre>body { background-color: lavender; }</pre>
클래스 선택자 : .	<pre>.title { font-size: 45px; }</pre>
아이디 선택자 : #	<pre>#menu1 { color: pink; }</pre>
특성 선택자 : []	<pre>img[src="./backgroundimg.jpg"] { width: 1000px; }</pre>
다중선택자 : a, b	<pre>h3, h4 { color: yellowgreen; }</pre>

| CSS 구성 - 주석

Ctrl + /

```
/* 주식입니다! */
```

```
h3, h4 {  
    color: yellowgreen;  
}
```

```
/* ~~~ */
```

| CSS 특성

1. 우선순위

2. 폭포수

3. 상속

| CSS 특성 - 우선순위

1. !important
2. 인라인 스타일
3. id 선택자
4. class 선택자
5. 태그 선택자
6. 전체 선택자
7. 상속

단! 같은 순위에서는 문서상으로 가장 나중에 입력된 속성이 적용

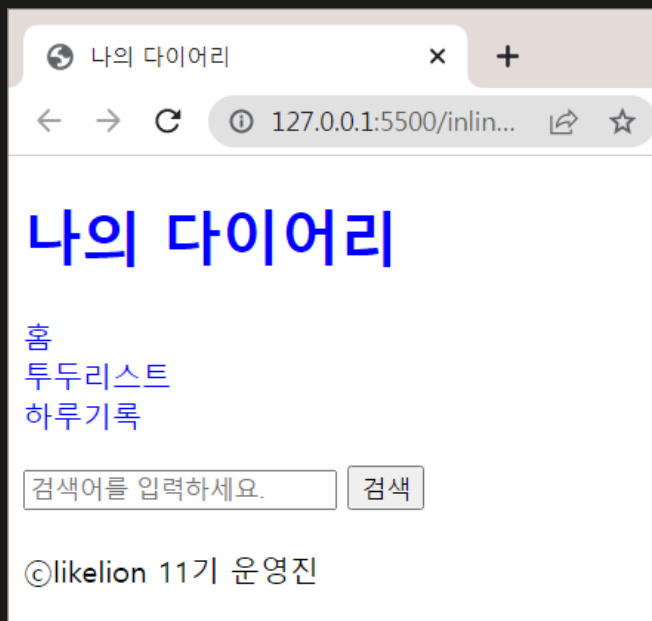
CSS 특성 - 우선순위

!important

이전에 지정된 스타일을 무시하고 가장 우선적으로 적용

(강제적인 특성이라 되도록 지양하는 것이 좋음.)

```
h1, p {  
  color : blue !important;  
}  
  
p {  
  color: yellowgreen;  
}
```



CSS 특성 - 상속

부모 요소의 속성을 자식 요소도 물려 받는 것.
(모두 상속되는 것은 아님.)

```
h1 {  
  color : blue;  
}
```

```
nav {  
  color: hotpink;  
}
```

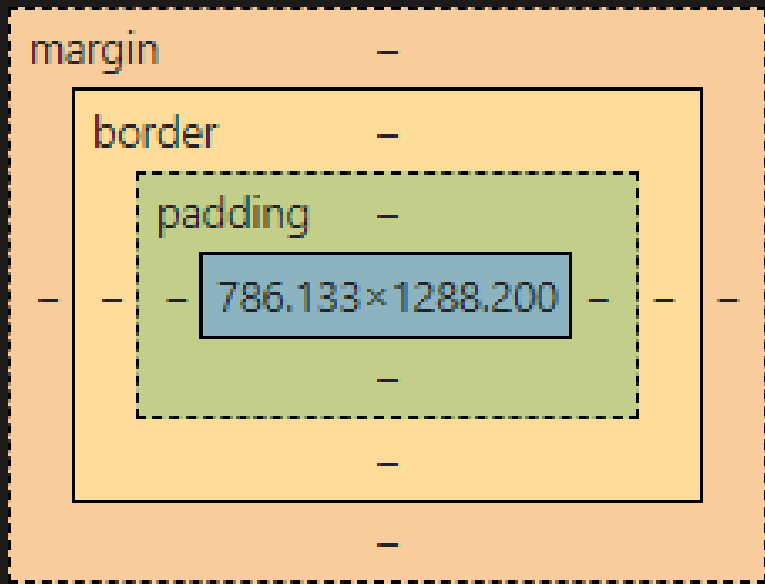
```
<nav>  
  <p>  
    홈<br/>  
    투두리스트<br/>  
    하루기록<br/>  
  </p>  
</nav>
```

나의 다이어리

홈
투두리스트
하루기록

©likelion 11기 운영진

Box Model



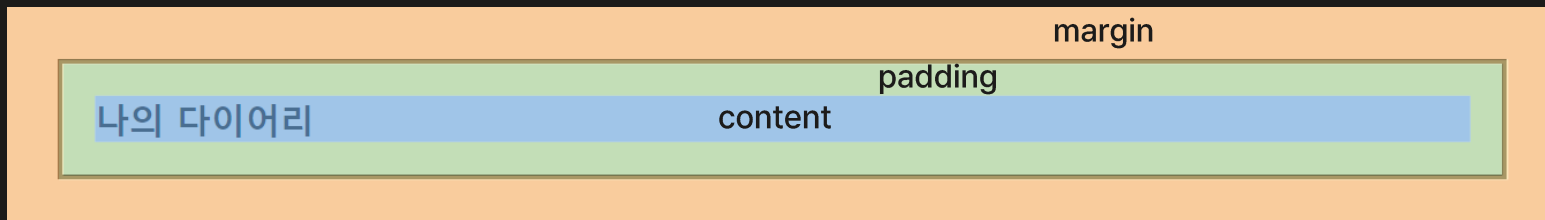
margin : 태그 외부의 여백

border : 테두리

padding : content와 테두리 사이의 여백

content : 글, 이미지 등 실제 내용이 차지하는 부분

| Box Model



```
h1 {  
  margin: 50px;  
  padding: 30px;  
  border: 5px solid black;  
}
```

개발자도구 F12 로 확인 가능!

| Box Model – padding & margin

1. 네 면 모두 같은 사이즈로 적용

`margin : 3px;`

2. 위아래 / 양옆 같은 사이즈로 적용

`margin : 5px(위아래) 10px(양옆);`

3. 네 면 모두 각각 적용

`margin : 5px(위) 3px(오른쪽) 10px(아래) 7px(왼쪽);`

`margin-top, margin-bottom, margin-left, margin-right`로도 적용 가능
padding도 똑같이 적용!

| Box Model – width & height

1. width : 요소의 너비

width : 100px;

2. height : 요소의 높이

height : 50px;

3. 콘텐츠 영역을 대상으로 함

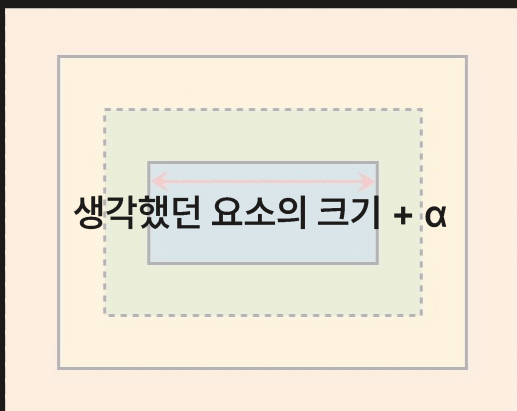
전체 너비 : width + left padding + right padding + left border + right border
+ left margin + right margin

전체 높이 : height + top padding + bottom padding + top border + bottom
border + top margin + bottom margin

Box Sizing 박스에 적용된 사이즈의 기준

content-box

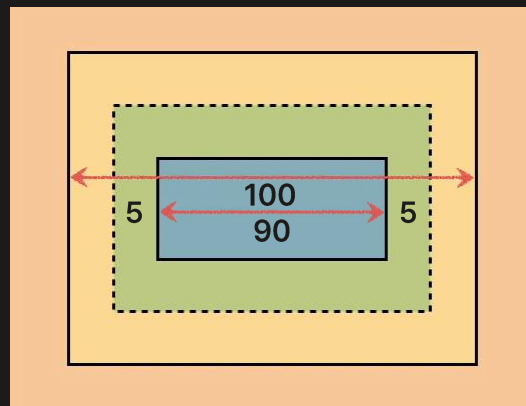
```
body {  
  box-sizing: content-box;  
  width: 100px;  
  padding: 5px;  
}
```



요소 사이즈에 패딩과 테두리 포함 X

border-box

```
body {  
  box-sizing: border-box;  
  width: 100px;  
  padding: 5px;  
}
```



요소 사이즈에 패딩과 테두리 포함 O

| Text 속성

1. 글자 색상

color : red / rgb(153,123,253) / #색상코드

2. 글자 크기

font-size : 24px / large / medium / small

3. 글씨체

font-family : 폰트이름

4. 글자 굵기

font-weight : normal(기본) / 500(굵자) / bold / lighter

5. 글자 정렬

font-align : center / left / right 등

| Border 속성

```
border : width style color;
```

1. 선 두께

`border-width : 2px ;`

2. 선 스타일

`border-style : dotted / dashed / solid / none ;`

3. 선 색상

`border-color : red / rgb(153,123,253) / #색상코드`

4. 테두리 모서리

`border-radius : 5px (개별적으로 지정 가능)`

| Border

```
input {  
  border: 0px;  
  border-bottom: 2px solid #d7d7d7;  
  width: 375px;  
  height: 45px;  
  font-size: 16px;  
  background-color: #f5f5f5;  
}
```

나의 다이어리

검색어를 입력하세요

홈

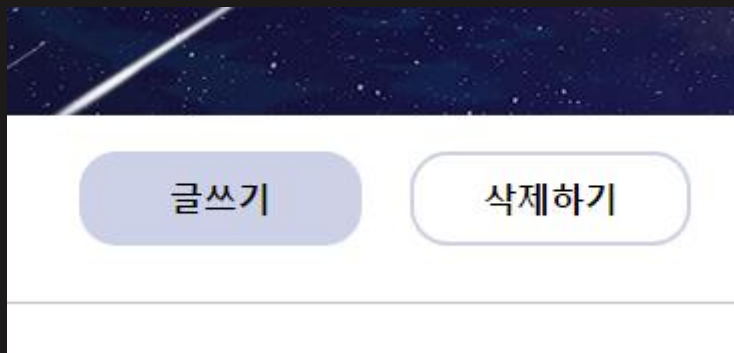
투두리스트

하루기록

| Border

```
.writebtn {  
  width: 125px;  
  height: 40px;  
  margin: 15px 5px 0 45px;  
  border: 0px;  
  border-radius: 10px;  
  background-color: #CCD1E7;  
  font-size: 16px;  
  font-weight: bold;  
}
```

```
.deletebtn {  
  width: 125px;  
  height: 40px;  
  margin: 15px;  
  border: 3px solid #CCD1E7;  
  border-radius: 10px;  
  background-color: white;  
  font-size: 16px;  
  font-weight: bold;  
}
```



| Border

```
a {  
  text-decoration: none;  
  color: black;  
}  
a:link {  
  color: black;  
}  
a:visited{  
  color: #CCD1E7;  
}  
a:hover {  
  color: purple;  
}
```

a:link : 방문 전 링크 상태

a:visited : 방문 후 링크 상태

a:hover : 마우스 오버 시 링크 상태



| Animation 속성

1. **animation-name** : 애니메이션 이름
 - 이름을 설정해야 애니메이션 재생 가능
2. **animation-duration** : 애니메이션 재생 시간
 - 0 or 음수 -> 재생되지 않음
3. **animation-delay** : 애니메이션 시작 지연 시간
 - 애니메이션 시작을 지연할 시간 설정 (0 / now / 숫자단위)
4. **animation-direction** : 애니메이션 재생 방향
 - normal / alternate / reverse / alternate-reverse
5. **animation-iteration-count** : 애니메이션 재생 횟수
 - 기본값은 1, 설정횟수만큼 재생, infinite는 무한 반복

| Animation 속성

```
.writebtn {  
    /* animation 연습 */  
    animation-name: BtnDown;  
    animation-duration: 0.3s;  
    animation-direction: normal;  
    animation-iteration-count: 1;  
    animation-fill-mode: forwards;  
}  
.writebtn:hover {  
    /* animation 연습 */  
    animation-name: BtnUp;  
    animation-duration: 0.3s;  
    animation-direction: normal;  
    animation-iteration-count: 1;  
    animation-fill-mode: forwards;  
    cursor: pointer;  
}
```

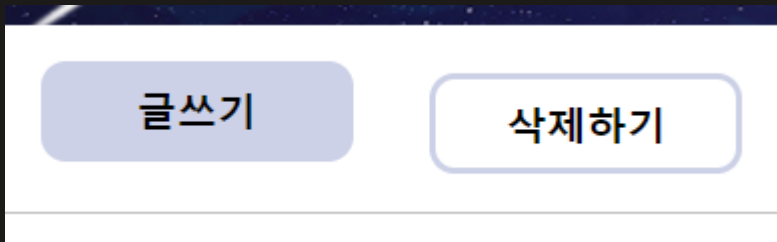


animation-fill-mode : 애니메이션 상태 제어

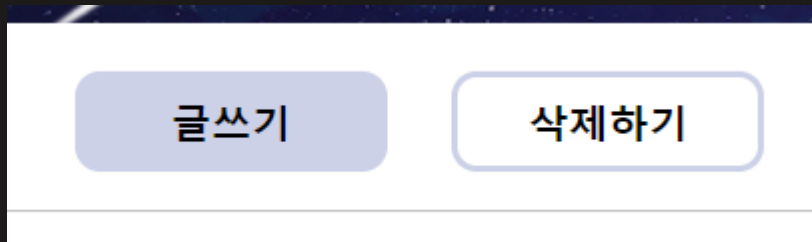
| Animation 속성 _@keyframes

애니메이션을 재생할 각 프레임의 스타일을 정의하는 것

```
@keyframes BtnUp {  
  0%{  
    position: relative;  
    top:0px;  
    left:0px;  
  }  
  100%{  
    position: relative;  
    top: -5px;  
    left: -5px;  
  }  
}
```



```
@keyframes BtnDown {  
  0%{  
    position: relative;  
    top:-5px;  
    left:-5px;  
  }  
  100%{  
    position: relative;  
    top: 0px;  
    left: 0px;  
  }  
}
```



| Transition 속성

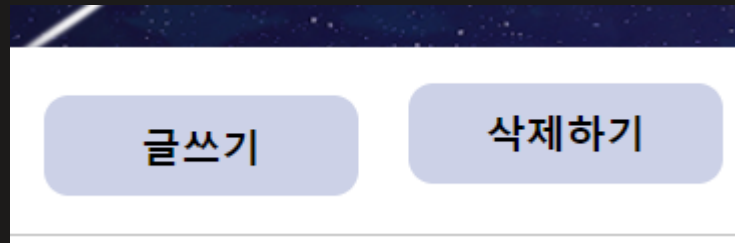
* transition은 자동으로 발동되지 않는다

transition : property duration function delay;

1. transition-property : css 프로퍼티명 지정
 - 트랜지션의 대상이 되는 css 프로퍼티
2. transition-duration : 트랜지션 지속시간
 - 기본값 0s -> 트랜지션 효과 x / 프로퍼티와 일대일 대응
3. transition-timing-function: 효과의 변화 리듬
 - ease / linear / ease-in / ease-out / ease-in-out
4. transition-delay : 트랜지션 시작 지연 시간
 - 일정 대기 시간 후에 트랜지션 효과 실행

Transition 속성

```
.deletebtn:hover {  
  background-color: #CCD1E7;  
  position: relative;  
  top: -5px;  
  left: -5px;  
  
  /* hover에서만발동 */  
  transition-property: background-color, top, left;  
  transition-duration: 0.3s, 0.2s, 0.2s;  
  transition-timing-function: ease, ease-in-out, ease-in;  
  transition-delay: 0,0,0;  
  
  transition: all 0.3s ease 0s;  
}
```



과제

html 시간에 만든 자신의 블로그에
오늘 배운 css 속성 모두 적용시켜
보기

감사합니다