



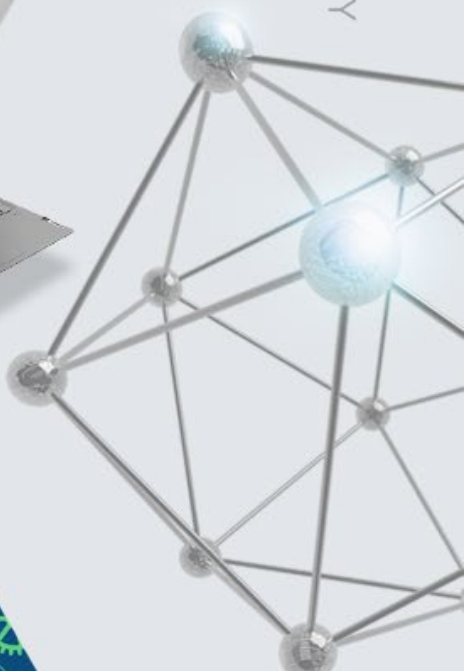
한국기술교육대학교  
온라인평생교육원

The 4th Industrial Revolution is characterized by super connectivity and super intelligence, where various products and services are connected to the network, and artificial intelligence and information communication technologies are used in 3D printing, unmanned transportation, robotics, Of the world's most advanced technologies.

T  
E  
C  
H  
N  
O  
L  
O  
G  
Y

# 웹 표준에 맞는 HTML5 프로그래밍

## UI 설계 확인하기



The 4th Industrial Revolution is characterized by super connectivity and super intelligence, where various products and services are connected to the network, and artificial intelligence and information communication technologies are used in 3D printing, unmanned transportation, robotics, Of the world's most advanced technologies.



# UI 설계 확인하기



## 학/습/목/표

1. 이미지 슬라이드를 CSS와 JavaScript로 구현할 수 있다.
2. 이미지 페이드 인-아웃을 CSS와 JavaScript로 구현할 수 있다.



## 학/습/내/용

1. 이미지 슬라이드(좌우)
2. 이미지 슬라이드(상하)
3. 페이드 인-아웃 구현하기

### 1. 이미지 슬라이드(좌우)

#### 1) CSS로 구현하기

##### (1) HTML 작성하기

```
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Automatical Slide In Horizontly</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div class="wrap">
      <div class="slide">
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

→ CSS 스타일 시트 사용

→ wrap : 전체를 감싸는 와이어프레임의 틀을 지정(사용자의 편의에 따라 자유롭게 지정 가능)

→ alt : 이미지에 대한 설명, 오류에 대한 대비

### 1. 이미지 슬라이드(좌우)

#### 1) CSS로 구현하기

##### (2) CSS 작성하기

```
* {  
  margin: 0 auto;  
  padding: 0;  
}  
  
ul,  
li {  
  list-style: none;  
}  
  
.wrap {  
  width: 503px;  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
}  
  
.slide {  
  height: 300px;  
  overflow: hidden;  
}  
  
.slide ul {  
  width: calc(100% * 4);  
  display: flex;  
  -webkit-animation: slide 8s infinite;  
  animation: slide 8s infinite;  
}  
  
.slide li {  
  width: calc(100% / 4);  
  height: 300px;  
  text-align: center;  
}
```

1/2

→ 브라우저의 default값 초기화

→ 이미지의 전체 사이즈가 슬라이드 레이아웃의 크기를 벗어났을 때  
벗어난 부분은 보이지 않게 하는 작업

### 1. 이미지 슬라이드(좌우)

#### 1) CSS로 구현하기

##### (2) CSS 작성하기

```
@keyframes slide {  
  0% { margin-left: 0; }  
  10% { margin-left: 0; }  
  25% { margin-left: -100%; }  
  35% { margin-left: -100%; }  
  50% { margin-left: -200%; }  
  60% { margin-left: -200%; }  
  75% { margin-left: -300%; }  
  85% { margin-left: -300%; }  
  100% { margin-left: 0; }  
}
```

2/2

→ slide : 사용자가 지정한 이름

→ 반복적으로 슬라이드 쇼가 보이게 하는 작업

##### (3) CSS Syntax

```
@keyframes animationname {keyframes-selector {css-styles;}}  
animation: name duration timing-function delay iteration-count  
direction fill-mode play-state;
```

→ keyframes를 통한 애니메이션 작성 방법

### 1. 이미지 슬라이드(좌우)

#### 1) CSS로 구현하기

##### 실행 결과



[ 슬라이드 종료 ]



[ 슬라이드 시작 ]



[ 슬라이드 3 ]



[ 슬라이드 1 ]



[ 슬라이드 2 ]



### 1. 이미지 슬라이드(좌우)

#### 2) JavaScript로 구현하기

##### (1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Automatic Slide Show1</title>
    <link rel="stylesheet" href="css/style.css" />
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.
min.js"></script>
    <script src="script/script.js"></script>
  </head>
  <body>
    <div class="contents">
      <h3>Automatic Slide Show1</h3>
      <div class="banner">
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

→ stylesheet 필요

→ 사용자가 작성한 JavaScript 필요

### 1. 이미지 슬라이드(좌우)

#### 2) JavaScript로 구현하기

##### (2) CSS 작성하기

```
body {  
  font-size: 11pt;  
  padding: 0;  
  margin: 0;  
  text-align: center;  
}  
h3 {  
  color: #85144b;  
  font-size: 14pt;  
  margin: 2 auto;  
  padding: 2px;  
}  
.contents {  
  width: 400px;  
  height: 280px;  
  background-color: #d6d6d6;  
  margin: 0 auto;  
}  
.banner {  
  position: relative;  
  width: 340px;  
  height: 210px;  
  top: 1px;  
  margin: 0 auto;  
  padding: 0;  
  overflow: hidden;  
}
```

1/2

```
.banner ul {  
  position: absolute;  
  margin: 0px;  
  padding: 0;  
  list-style: none;  
}  
.banner ul li {  
  float: left;  
  width: 340px;  
  height: 210px;  
  margin: 0;  
  padding: 0;  
}
```

2/2

- body에 초기화 설정
- 이미지의 전체 사이즈가 슬라이드 레이아웃의 크기를 벗어났을 때 벗어난 부분은 보이지 않게 하는 작업
- 정확한 좌우 이동을 위한 이미지 배치 작업



### 1. 이미지 슬라이드(좌우)

#### 2) JavaScript로 구현하기

##### (3) JavaScript 작성하기

```
$(document).ready(function () {  
    var $banner = $(".banner").find("ul");  
    var $bannerWidth = $banner.children().outerWidth();  
    var $bannerHeight = $banner.children().outerHeight();  
    var $bannerLength = $banner.children().length;  
    var rollingId;  
    rollingId = setInterval(function () { rollingStart(); },  
2000);  
    function rollingStart() {  
        $banner.css("width", $bannerWidth * $bannerLength + "px");  
        $banner.animate({  
            left: -$bannerWidth + "px" }, 2500, function () {  
                $(this).append("<li>" + $(this).find("li:first").html()  
+ "</li>");  
                $(this).find("li:first").remove();  
                $(this).css("left", 0);  
            });  
        }  
    });  
});
```

→ 이미지의 길이가 아닌 이미지 개수

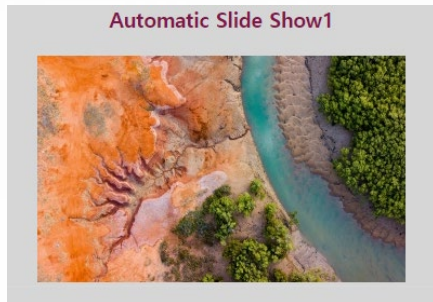
→ 이미지 슬라이드 쇼의 지속을 위한 변수

→ 1/1000초 단위 (2000=2초)

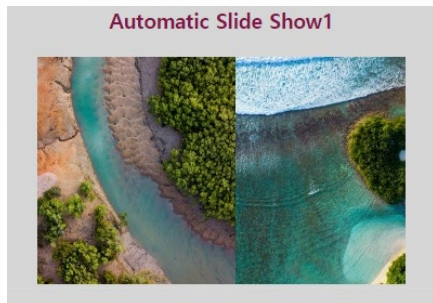
### 1. 이미지 슬라이드(좌우)

#### 2) JavaScript로 구현하기

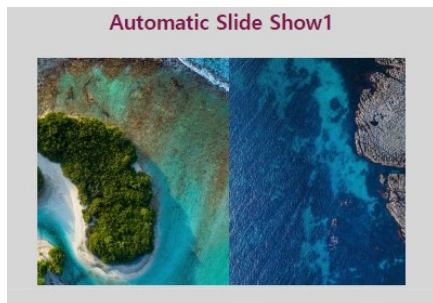
##### 실행 결과



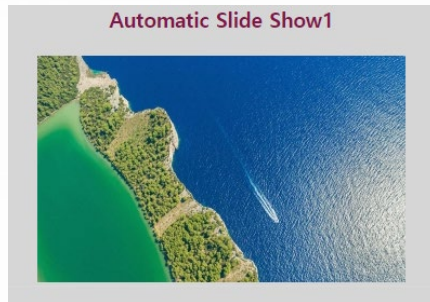
[ 슬라이드 시작 ]



[ 슬라이드 1 ]



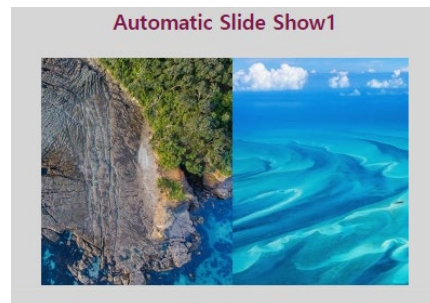
[ 슬라이드 2 ]



[ 슬라이드 종료 ]



[ 슬라이드 4 ]



[ 슬라이드 3 ]

## 2. 이미지 슬라이드(상하)

### 1) CSS로 구현하기

#### (1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Automatical Slide In Vertically</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div class="wrap">
      <div class="slide">
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

## 2. 이미지 슬라이드(상하)

### 1) CSS로 구현하기

#### (2) CSS 작성하기

```
* {  
  margin: 0 auto;  
  padding: 0;  
}  
ul,  
li {  
  list-style: none;  
}  
.wrap {  
  width: 503px;  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
}  
.slide {  
  height: 300px;  
  overflow: hidden;  
}  
.slide ul {  
  height: calc(100% * 4);  
  -webkit-animation: slide 8s infinite;  
  animation: slide 8s infinite;  
}  
.slide li {  
  height: calc(100% / 4);  
}
```

1/2

→ 좌우 이미지 슬라이드와의 차이점 : width에서 height로 슬라이드  
사이즈 변경



## 2. 이미지 슬라이드(상하)

### 1) CSS로 구현하기

#### (2) CSS 작성하기

```
@keyframes slide {  
  0% { margin-top: 0; }  
  10% { margin-top: 0; }  
  25% { margin-top: -300px; }  
  35% { margin-top: -300px; }  
  50% { margin-top: -600px; }  
  60% { margin-top: -600px; }  
  75% { margin-top: -900px; }  
  85% { margin-top: -900px; }  
  100% { margin-top: 0; }  
}
```

2/2

→ 슬라이드가 상하로 움직이도록 margin 변경

## 2. 이미지 슬라이드(상하)

### 1) CSS로 구현하기

#### 실행 결과



[ 슬라이드 종료 ]



[ 슬라이드 시작 ]



[ 슬라이드 3 ]



[ 슬라이드 1 ]



[ 슬라이드 2 ]



## 2. 이미지 슬라이드(상하)

### 2) JavaScript로 구현하기

#### (1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Automatic Slide Show2</title>
    <link rel="stylesheet" href="css/style.css" />
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.
min.js"></script>
    <script src="script/script.js"></script>
  </head>
  <body>
    <div class="contents">
      <h3>Automatic Slide Show2</h3>
      <div class="banner">
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

## 2. 이미지 슬라이드(상하)

### 2) JavaScript로 구현하기

#### (2) CSS 작성하기

```
body {
  font-size: 11pt;
  padding: 0;
  margin: 0;
  text-align: center;
}
h3 {
  color: #85144b;
  font-size: 14pt;
  margin: 2 auto;
  padding: 2px;
}
.contents {
  width: 400px;
  height: 280px;
  background-color: #d6d6d6;
  margin: 0 auto;
}
.banner {
  position: relative;
  width: 340px;
  height: 210px;
  top: 1px;
  margin: 0 auto;
  padding: 0;
  overflow: hidden;
}
```

1/2

```
.banner ul {
  position: absolute;
  margin: 0px;
  padding: 0;
  list-style: none;
}
.banner ul li {
  width: 340px;
  height: 210px;
  margin: 0;
  padding: 0;
}
```

2/2



## 2. 이미지 슬라이드(상하)

### 2) JavaScript로 구현하기

#### (3) JavaScript 작성하기

```
$(document).ready(function () {  
    var $banner = $(".banner").find("ul");  
    var $bannerWidth = $banner.children().outerWidth();  
    var $bannerHeight = $banner.children().outerHeight();  
    var $bannerLength = $banner.children().length;  
    var rollingId;  
    rollingId = setInterval(function () { rollingStart(); },  
2000);  
    function rollingStart() {  
        $banner.css("height", $bannerHeight + "px");  
        $banner.animate({  
            top: -$bannerHeight + "px" }, 1500, function () {  
                $(this).append("<li>" + $(this).find("li:first").html()  
+ "</li>");  
                $(this).find("li:first").remove();  
                $(this).css("top", 0);  
            });  
        }  
    });  
});
```

→ 좌우 이미지 슬라이드와 JavaScript에서의 차이점

## 2. 이미지 슬라이드(상하)

### 2) JavaScript로 구현하기

#### 실행 결과



[ 슬라이드 시작 ]



[ 슬라이드 1 ]



[ 슬라이드 2 ]



[ 슬라이드 종료 ]



[ 슬라이드 4 ]



[ 슬라이드 3 ]

### 3. 페이드 인-아웃 구현하기

#### 1) CSS로 구현하기

##### (1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Fade-In & Fade-Out Automatically</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div class="wrap">
      <div class="slide">
        <ul>
          <li></li>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

### 3. 페이드 인-아웃 구현하기

#### 1) CSS로 구현하기

#### (2) CSS 작성하기

```
* {  
  margin: 0 auto;  
  padding: 0;  
}  
ul,  
li {  
  list-style: none;  
}  
.wrap {  
  width: 503px;  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
}  
.slide {  
  height: 300px;  
  overflow: hidden;  
}  
.slide ul {  
  position: relative;  
  height: 100%;  
}
```

1/2



### 3. 페이드 인-아웃 구현하기

#### 1) CSS로 구현하기

##### (2) CSS 작성하기

```
.slide li {  
  position: absolute;  
  left: 0;  
  right: 0;  
  top: 0;  
  bottom: 0;  
  opacity: 0;  
  animation: fade 8s infinite;  
}  
.slide li:nth-child(1) { animation-delay: 0s }  
.slide li:nth-child(2) { animation-delay: 2s }  
.slide li:nth-child(3) { animation-delay: 4s }  
.slide li:nth-child(4) { animation-delay: 6s }  
@keyframes fade {  
  0% { opacity: 0; }  
  5% { opacity: 1; }  
  25% { opacity: 1; }  
  30% { opacity: 0; }  
  100% { opacity: 0; }  
}
```

2/2

- li 태그의 하위에 포함되는 list들을 nth-child(n)으로 지정하여 애니메이션 효과를 지정
- 선명도를 조정한 페이드 인-아웃 효과를 설정

### 3. 페이드 인-아웃 구현하기

#### 1) CSS로 구현하기

##### 실행 결과



[ 페이드 인-아웃 시작 ]



[페이드 인-아웃 1]



[페이드 인-아웃 2]



[페이드 인-아웃 종료]

### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

##### (1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>Automatic Fade-In & Fade-Out</title>
    <link href="css/style.css" type="text/css" rel="stylesheet" />
    <script src="script/jquery-1.12.4.js"></script>
    <script src="script/script.js" defer
type="text/javascript"></script>
  </head>
  <body>
    <div class="slideshow-container">
      <div class="mySlides fade">
        <div class="numbertext">1 / 3</div>
        
      </div>
      <div class="mySlides fade">
        <div class="numbertext">2 / 3</div>
        
      </div>
    </div>
    <br />
    <div style="text-align: center;">
      <span class="dot"></span>
      <span class="dot"></span>
      <span class="dot"></span>
    </div>
  </body>
</html>
```

- 사용자가 작성한 JavaScript 필요
- 슬라이드 이미지에 dot가 보이도록 설정

### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

##### (2) CSS 작성하기

```
* {  
  box-sizing: border-box;  
}  
body {  
  font-family: Verdana, sans-serif;  
}  
.mySlides {  
  display: none;  
}  
img {  
  vertical-align: middle;  
}  
.slideshow-container {  
  max-width: 1000px;  
  position: relative;  
  margin: auto;  
}  
.text {  
  color: #f2f2f2;  
  font-size: 15px;  
  padding: 8px 12px;  
  position: absolute;  
  bottom: 8px;  
  width: 100%;  
  text-align: center;  
}
```

1/3

### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

##### (2) CSS 작성하기

```
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
.active {
  background-color: #717171;
}
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}
```

2/3

→ CSS로 fade에 대한 애니메이션을 선언

### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

##### (2) CSS 작성하기

```
@-webkit-keyframes fade {  
  from {  
    opacity: 0.4;  
  }  
  to {  
    opacity: 1;  
  }  
}  
@keyframes fade {  
  from {  
    opacity: 0.4;  
  }  
  to {  
    opacity: 1;  
  }  
}
```

3/3



### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

##### (1) JavaScript 작성하기

```
var slideIndex = 0;
showSlides();
function showSlides() {
    var i;
    var slides = document.getElementsByClassName('mySlides');
    var dots = document.getElementsByClassName('dot');
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = 'none';
    }
    slideIndex++;
    if (slideIndex > slides.length) {
        slideIndex = 1;
    }
    for (i = 0; i < dots.length; i++) {
        dots[i].className = dots[i].className.replace(' active',
        '');
    }
    slides[slideIndex - 1].style.display = 'block';
    dots[slideIndex - 1].className += ' active';
    setTimeout(showSlides, 2000);
}
```

→ 반복을 위한 for문

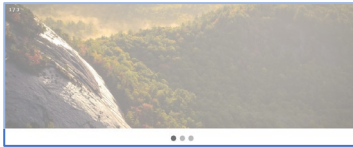
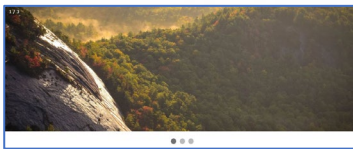
→ dot를 위한 for문



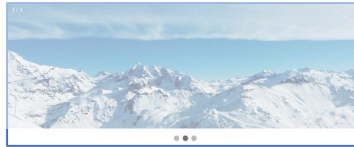
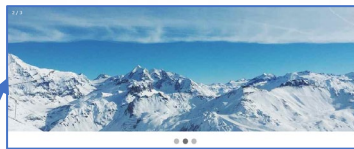
### 3. 페이드 인-아웃 구현하기

#### 2) JavaScript로 구현하기

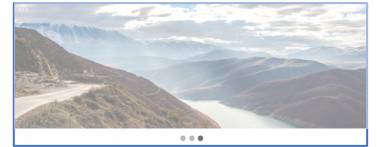
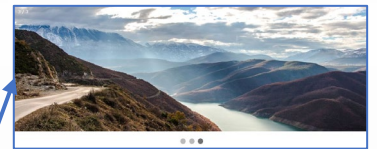
##### 실행 결과



[ 페이드 인-아웃  
시작 ]



[페이드 인-아웃  
1 ]



[페이드 인-아웃  
종료 ]



## 1. 이미지 슬라이드(좌우)

- CSS로 구현하기
  - @keyframes 를 이용함
  - margin-left 로 좌측 좌표를 선언하여 좌우 슬라이드를 구현할 수 있음
- JavaScript로 구현하기
  - 지정 시간마다 반복되는 setInterval() 함수를 활용함
  - animate 메소드를 통해 지정 시간 마다 요소의 left 속성을 할당하여 구현할 수 있음

## 2. 이미지 슬라이드(상하)

- CSS로 구현하기
  - @keyframes 를 이용함
  - margin-top 로 상단 좌표를 선언하여 상하 슬라이드를 구현할 수 있음
- JavaScript로 구현하기
  - 지정 시간마다 반복되는 setInterval() 함수를 활용함
  - animate 메소드를 통해 지정 시간 마다 요소의 top 속성을 할당하여 구현할 수 있음



### 3. 페이드 인-아웃 구현하기

- CSS로 구현하기
  - @keyframes 를 이용함
  - opacity 로 투명도를 선언하여 페이드 인-아웃을 구현할 수 있음
- JavaScript로 구현하기
  - 지정 시간마다 호출하는 setTimeout() 함수를 활용함
  - style.display 속성을 'none', 'block'으로 할당하여 페이드 인-아웃을 구현할 수 있음