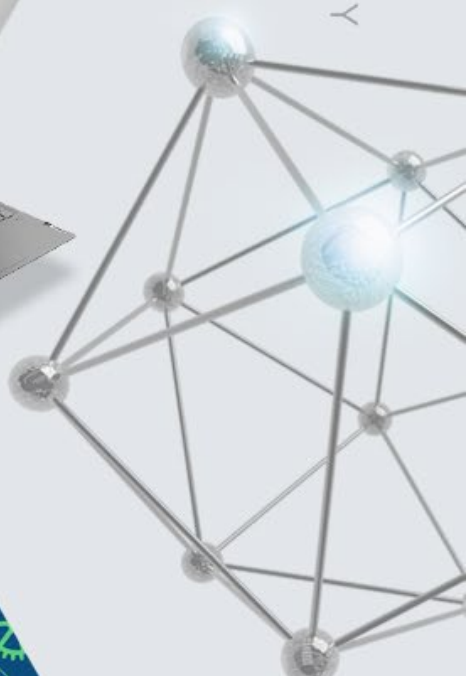
한국기술교육대학교
온라인평생교육원

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.

TECHNOLOGY

웹 표준에 맞는 HTML5 프로그래밍

반응형 웹(RWD) 화면 구현하기



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.

반응형 웹(RWD) 화면 구현하기

학/습/목/표

1. Float를 활용하여 화면을 구현할 수 있다.
2. W3.CSS를 활용하여 화면을 구현할 수 있다.
3. Grid를 활용하여 화면을 구현할 수 있다.
4. Media query를 활용하여 Responsive한 화면을 구현할 수 있다.

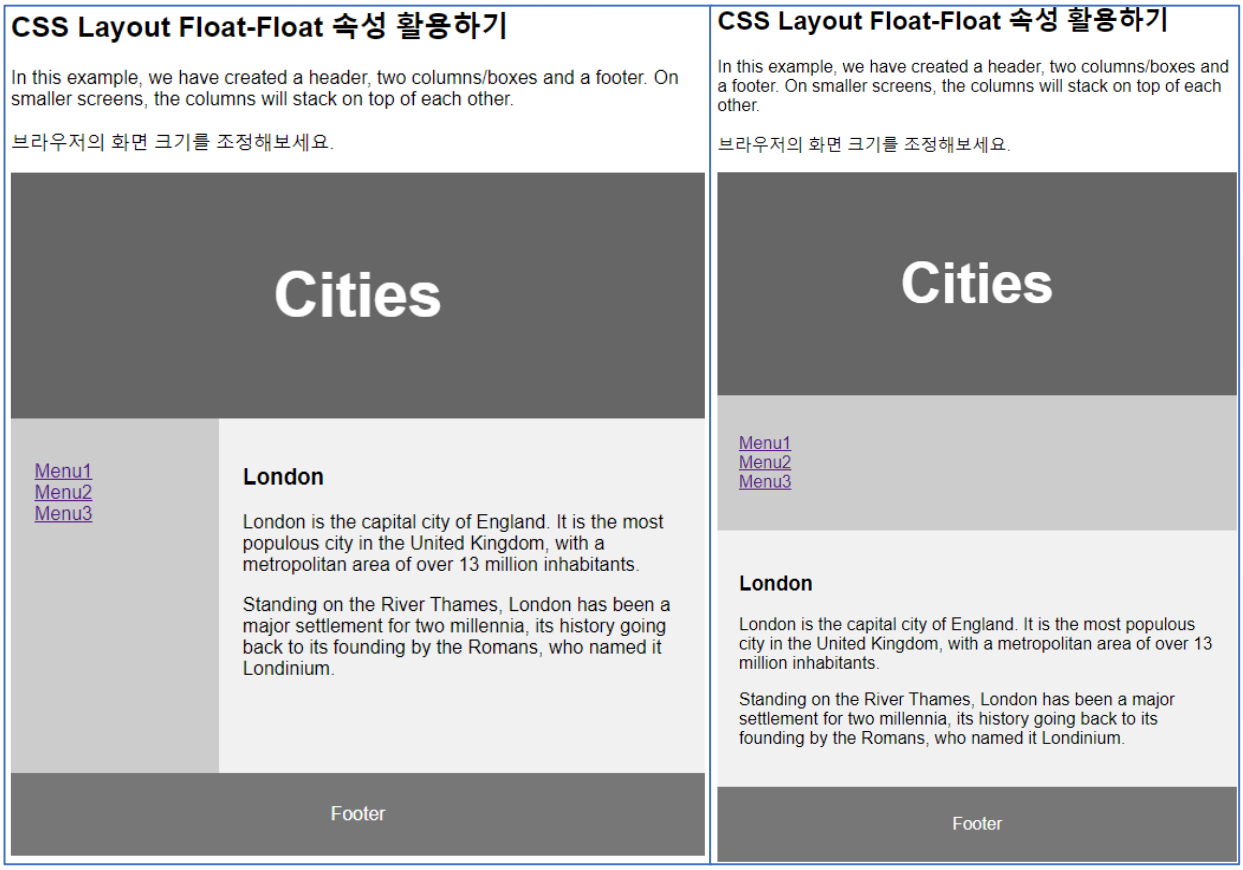
학/습/내/용

1. Float 활용 화면 구현
2. W3.CSS 활용 화면 구현
3. Grid 활용 화면 구현
4. Media Query 활용 화면 구현

1. Float 활용 화면 구현

1) CSS Layout Float 실행결과

(1) CSS Layout Float 실행결과



1. Float 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <title>CSS Template</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <h2>CSS Layout Float-Float 속성 활용하기</h2>
    <p>In this example, we have created a header, two
columns/boxes and a footer. On smaller screens,
       the columns will stack on top of each other.</p>
    <p>브라우저의 화면 크기를 조정해보세요.</p>
    <header>
      <h2>Cities</h2>
    </header>
    <section>
      <nav>
        <ul>
          <li><a href="#">Menu1</a></li>
          <li><a href="#">Menu2</a></li>
          <li><a href="#">Menu3</a></li>
        </ul>
      </nav>
```

1/2

→ section안에 nav와 article로 나눔

1. Float 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
<article>
        <h1>London</h1>
        <p>London is the capital city of England. It is the
most populous city in the United Kingdom,
           with a metropolitan area of over 13 million
inhabitants.</p>
        <p>Standing on the River Thames, London has been a
major settlement for two millennia,
           its history going back to its founding by the
Romans, who named it Londinium.</p>
 </article>
</section>
 <footer>
   <p>Footer</p>
 </footer>
</body>
</html>
```

2/2

1. Float 활용 화면 구현

2) CSS 작성하기

(1) CSS 작성하기

```
* { box-sizing: border-box; }
body { font-family: Arial, Helvetica, sans-serif; }
header {
  background-color: #666;
  padding: 30px;
  text-align: center;
  font-size: 35px;
  color: white;
}
nav {
  float: left;
  width: 30%;
  height: 300px; /* only for demonstration, should be removed
*/
  background: #ccc;
  padding: 20px;
}
nav ul {
  list-style-type: none;
  padding: 0;
}
```

1/2

→ nav를 왼쪽 정렬. 전체 넓이의 30%로 지정

1. Float 활용 화면 구현

2) CSS 작성하기

(1) CSS 작성하기

```
article {
  float: left;
  padding: 20px;
  width: 70%;
  background-color: #f1f1f1;
  height: 300px; /* only for demonstration, should be removed */
}
section:after {
  content: "";
  display: table;
  clear: both;
}
footer {
  background-color: #777;
  padding: 10px;
  text-align: center;
  color: white;
}
@media (max-width: 600px) {
  nav, article {
    width: 100%;
    height: auto;
  }
}
```

2/2

→ article을 왼쪽 정렬. 전체 넓이의 70%로 지정

→ float로 인한 오류를 위한 설정

→ 반응형 웹을 구현

2. W3.CSS 활용 화면 구현

1) W3.CSS Layout 실행결과

(1) W3.CSS Layout 실행결과

- W3 easy alignment

Hello W3.CSS Layout. Hello W3.CSS Layout. Hello W3.CSS Layout. Hello W3.CSS Layout.	Hello W3.CSS Layout.	Hello W3.CSS Layout.
[W3-cell-top (default)]	[W3-cell-middle]	[W3-cell-bottom]

https://www.w3schools.com/w3css/w3css_layout.asp

2. W3.CSS 활용 화면 구현

1) W3.CSS Layout 실행결과

(1) W3.CSS Layout 실행결과

- W3 CSS Colors

Red	Pink
Purple	Deep Purple
Indigo	Blue
Light Blue	Cyan
Aqua	Teal
Green	Light Green
Lime	Sand
Khaki	Yellow
Amber	Orange
Deep Orange	Blue Gray
Brown	Light Gray
Gray	Dark Gray
Pale Red	Pale Yellow
Pale Green	Pale Blue

https://www.w3schools.com/w3css/w3css_colors.asp

2. W3.CSS 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
<!DOCTYPE html>
<html>
  <title>W3.CSS Layout</title>
  <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
  <link rel="stylesheet" href="css/style.css" />
  <body>
    <h2>W3.CSS Layout</h2>
    <div class="w3-container w3-red w3-cell w3-cell-top">
      <p>Hello W3.CSS Layout.</p>
      <p>Hello W3.CSS Layout.</p>
      <p>Hello W3.CSS Layout.</p>
      <p>Hello W3.CSS Layout.</p>
    </div>
    <div class="w3-container w3-green w3-cell w3-cell-middle">
      <p>Hello W3.CSS Layout.</p>
    </div>
    <div class="w3-container w3-blue w3-cell w3-cell-bottom">
      <p>Hello W3.CSS Layout.</p>
    </div>
    <br>
```

1/5

→ W3.CSS로 레이아웃 지정

2. W3.CSS 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
<table class="w3-table">
  <caption>W3.CSS 색 이름</caption>
  <tr>
    <td class="w3-red" style="width: 50%;">
      <p>Red</p>
    </td>
    <td class="w3-pink" style="width: 50%;">
      <p>Pink</p>
    </td>
  </tr>
  <tr>
    <td class="w3-purple" style="width: 50%;">
      <p>Purple</p>
    </td>
    <td class="w3-deep-purple" style="width: 50%;">
      <p>Deep Purple</p>
    </td>
  </tr>
  <tr>
    <td class="w3-indigo" style="width: 50%;">
      <p>Indigo</p>
    </td>
    <td class="w3-blue" style="width: 50%;">
      <p>Blue</p>
    </td>
  </tr>
  <tr>
    <td class="w3-light-blue">
      <p>Light Blue</p>
    </td>
```

→ W3.CSS에 지정된 컬러이름 직접 사용

2/5

2. W3.CSS 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
      <td class="w3-cyan">
        <p>Cyan</p>
      </td>
    </tr>
    <tr>
      <td class="w3-aqua">
        <p>Aqua</p>
      </td>
      <td class="w3-teal">
        <p>Teal</p>
      </td>
    </tr>
    <tr>
      <td class="w3-green">
        <p>Green</p>
      </td>
      <td class="w3-light-green">
        <p>Light Green</p>
      </td>
    </tr>
    <tr>
      <td class="w3-lime">
        <p>Lime</p>
      </td>
      <td class="w3-sand">
        <p>Sand</p>
      </td>
    </tr>
    <tr>
      <td class="w3-khaki">
        <p>Khaki</p>
```

3/5

2. W3.CSS 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
      </td>
      <td class="w3-yellow">
        <p>Yellow</p>
      </td>
    </tr>
    <tr>
      <td class="w3-amber">
        <p>Amber</p>
      </td>
      <td class="w3-orange">
        <p>Orange</p>
      </td>
    </tr>
    <tr>
      <td class="w3-deep-orange">
        <p>Deep Orange</p>
      </td>
      <td class="w3-blue-gray">
        <p>Blue Gray</p>
      </td>
    </tr>
      <td class="w3-brown">
        <p>Brown</p>
      </td>
      <td class="w3-light-gray">
        <p>Light Gray</p>
      </td>
    </tr>
```

4/5

2. W3.CSS 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
      <tr>
        <td class="w3-gray">
          <p>Gray</p>
        </td>
        <td class="w3-dark-gray">
          <p>Dark Gray</p>
        </td>
      </tr>
      <tr>
        <td class="w3-pale-red">
          <p>Pale Red</p>
        </td>
        <td class="w3-pale-yellow">
          <p>Pale Yellow</p>
        </td>
      </tr>
      <tr>
        <td class="w3-pale-green">
          <p>Pale Green</p>
        </td>
        <td class="w3-pale-blue">
          <p>Pale Blue</p>
        </td>
      </tr>
    </table>
  </body>
</html>
```

5/5

2. W3.CSS 활용 화면 구현

3) CSS 작성하기

(1) CSS 작성하기

```
.w3-table {
  width: 600px;
  font-size: 14px;
}
p {
   text-align: center;
   margin: 0 0;
}
```

→ color table이 600px을 기준으로 반응형 웹에 적용 되도록 설정

3. Grid 활용 화면 구현

1) HTML 작성하기

(1) Grid 활용 화면 구현

- HTML 작성하기

```
<!DOCTYPE html>
<html>
  <head>
    <title>Grid Layout</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <h1>Grid Layout</h1>
    <p>This grid layout contains six columns and three rows:</p>
    <div class="grid-container">
      <div class="item1">Header</div>
      <div class="item2">Menu</div>
      <div class="item3">Main</div>
      <div class="item4">Right</div>
      <div class="item5">Footer</div>
    </div>
  </body>
</html>
```

→ div 태그를 사용해서 5개의 영역으로 구성

3. Grid 활용 화면 구현

2) CSS 작성하기

(1) CSS 작성하기

```
.item1 { grid-area: header; }
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }
.grid-container {
  display: grid;
  grid-template-areas:
    'header header header header header header'
    'menu main main main right right'
    'menu footer footer footer footer footer';
  grid-gap: 1px;
  background-color: #2196F3;
  padding: 5px;
}
.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
```

→ grid를 사용한 display를 선언

→ 그리드 영역에 지정된 이름만으로 화면 구성 가능

3) Grid 활용 화면 구현 실행결과

(1) Grid 활용 화면 구현 실행결과

실행 결과

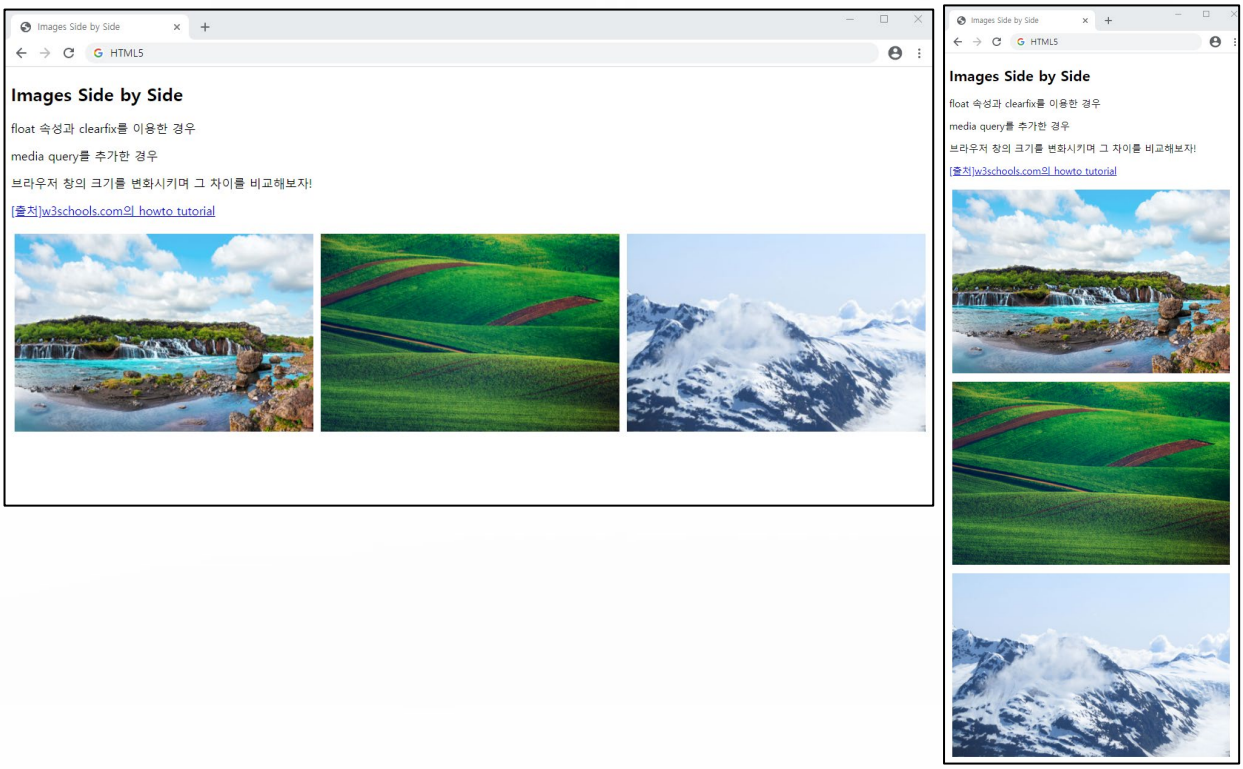
This grid layout contains six columns and three rows:

<table>
<tr><td colspan="3">Header</td></tr>
<tr><td rowspan="2">Menu</td><td>Main</td><td>Right</td></tr>
<tr><td colspan="2">Footer</td></tr>
</table>

4. Media Query 활용 화면 구현

1) 반응형 (Responsive) 화면 구현 실행결과

(1) 반응형 (Responsive) 화면 구현 실행결과



4. Media Query 활용 화면 구현

2) HTML 작성하기

(1) HTML 작성하기

```
<!DOCTYPE html>
<html>
  <head>
    <title>Images Side by Side</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <h2>Images Side by Side</h2>
    <p>float 속성과 clearfix를 이용한 경우</p>
    <p>media query를 추가한 경우</p>
    <p>브라우저 창의 크기를 변화시키며 그 차이를 비교해보자!</p>
    <p><a
href="https://www.w3schools.com/howto/tryit.asp?filename=tryhow_cs
s_images_side_by_side_resp">
      [출처]w3schools.com의 howto tutorial</a></p>
    <div class="row">
      <div class="column">
        <img src="images/img_5terre.jpg" alt="Terre" style="width:
100%;" />
      </div>
      <div class="column">
        <img src="images/img_forest.jpg" alt="Forest"
style="width: 100%;" />
      </div>
      <div class="column">
        <img src="images/img_mountains.jpg" alt="Mountains"
style="width: 100%;" />
      </div>
    </div>
  </body>
</html>
```

→ 이미지의 개수만큼 column생성

→ 장애우를 위한 설명 또는 이미지 파일 오류시 확인을 위한 설명

4. Media Query 활용 화면 구현

3) CSS 작성하기

(1) CSS 작성하기

```
* {
  box-sizing: border-box;
}
.column {
  float: left;
  width: 33.33%;
  padding: 5px;
}
/* Clearfix (clear floats) */
.row::after {
  content: "";
  clear: both;
  display: table;
}
/* Responsive layout */
@media screen and (max-width: 500px) {
  .column {
    width: 100%;
  }
}
```

→ float로 인한 오류를 방지하기 위한 초기화

→ 반응형 웹 구현을 위한 작업 : 디바이스 화면 크기 500px이하에서는 넓이를 100%로 구현하도록 설정

1. Float 활용 화면 구현

- CSS Layout Float 실행결과
- HTML 작성하기 : 화면 구성을 위한 영역 지정(header, section, nav, article, footer 등)
- CSS 작성하기 : @media를 사용한 반응형 웹 구현

2. W3.CSS 활용 화면 구현

- W3.CSS Layout 실행결과 : W3 easy alinment를 사용해서 텍스트의 위치와 색을 구현
- HTML 작성하기 : W3schools.com을 호출하기
  - W3-container, W3-cell, W3-color등
- CSS 작성하기 : 넓이, 색상등 이미 HTML에 구현된 것 외의 스타일링 지정하기

3. Grid 활용 화면 구현

- HTML 작성하기 : div 태그를 이용한 영역 구성, 각 영역이름 지정
- CSS 작성하기 : display: grid를 사용. 그리드 영역에 지정된 이름만으로 화면 구성
- Grid 활용 화면 구현 실행결과 : 별도의 코드없이도 영역 구현

4. Media query 활용 화면 구현

- 반응형 (Responsive) 화면 구현 실행결과 : 웹 페이지를 구성하는 영역이 화면의 크기에 따라 사용자의 편의에 맞게 변함
- HTML 작성하기
- CSS 작성하기 : @media로 반응형 웹 구현. max-width를 사용하여 디바이스 크기 지정 가능