

15장 레이아웃

각 요소를 페이지에 배치하는 방법과 매력적인 페이지 레이아웃을 만드는 방법을 살펴본다.

- 정상적인 흐름, 상대 위치, 절대 위치, float 프로퍼티를 사용해 요소의 위치를 지정하는 방법
- 다양한 화면 크기와 해상도로 생산된 기기의 종류와 이러한 기기가 디자인 프로세스에 미치는 영향
- 고정 너비와 유동 레이아웃 간의 차이점과 생성 방법
- 그리드를 사용하여 페이지를 좀 더 체계적으로 작성하는 방법

요소 위치 지정의 핵심 개념

- 블록 구성 : CSS는 각 HTML 요소를 자체적인 박스 안에 있는 것처럼 처리한다. 이러한 박스는 **블록 레벨 박스** 또는 **인라인 박스**가 된다.
 - 블록 레벨 박스 : 새로운 줄에 시작
 - 인라인 박스 : 주변 텍스트 사이에 위치
- 박스의 너비와 높이를 설정하여 각 박스가 차지하는 여백을 제어
- 박스의 구분을 위해 테두리, 마진, 패딩, 배경색을 사용한다.
- 부모 요소 : 하나의 블록 레벨 요소가 또 다른 블록 레벨 요소 안에 위치할 경우 외부에 있는 박스를 **부모 요소** 또는 **포함하는 요소**라 한다.
 - `<div>` 요소나 다른 블록 요소를 사용해 각 요소를 **그룹화** 한다.
 - 박스는 다른 블록 레벨 요소 안에 **중첩** 될 수 있다.

요소의 위치 제어

CSS에는 정상적인 흐름, 상대 위치, 절대 위치 같이 페이지의 레이아웃을 제어할 수 있는 **위치 결정 방식**이 있다. 박스의 위치를 지정하는 경우 **박스 오프셋 프로퍼티**를 사용한다. 요소의 위치가 변하면서 겹칠 수 있는데 이때 **z-index 프로퍼티**를 이용해 겹쳐지는 순서를 지정할 수 있다.

위치 제어 프로퍼티

1. position
2. float

박스 오프셋 프로퍼티

1. top
2. bottom
3. left
4. right

- 위치 결정 방식
 - 정상적인 흐름 : 모든 블록 레벨 요소는 새로운 줄에 위치하고 페이지 아래쪽으로 차례로 나

탄난다.

- 상대 위치 : 정상적인 흐름의 위치를 기준으로 요소를 위쪽, 오른쪽, 아래쪽, 왼쪽 등으로 이동한다.(현재 요소 위치 기준)
- 절대 위치 : 부모 요소를 기준으로 요소를 배치한다. 사용자가 페이지를 위아래로 스크롤 하면 함께 이동한다.
- 고정 위치 : 브라우저 창을 기준으로 요소를 배치하는 절대 위치의 한 형태로 스크롤 하더라도 움직이지 않는다.
- 부유 요소 : 해당 요소를 정상적인 흐름에서 벗어나 포함하는 박스의 왼쪽이나 오른쪽에 떨어져 위치한다.

정상적인 흐름 `position:static`

각 블록 레벨 요소가 다음 요소의 위에 위치한다. 기본 방식으로 별도로 CSS프로퍼티를 지정할 필요가 없다.

```
position:static;
```

상대 위치 `position:relative`

요소가 정상적인 흐름에 있었을 경우의 위치를 기준으로 요소를 이동한다.

```
position:relative
```

- 작성방법
 1. `position` 프로퍼티 값을 `relative` 설정
 2. 박스 오프셋 프로퍼티로 위치 지정

```
<!DOCTYPE html>
<html>
  <head>
    <title>상대 위치</title>
    <style type="text/css">
      body {
        width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}
      p {
        width: 450px;}
      p.example {
        position: relative;
        top: 10px;
        left: 100px;}
    </style>
  </head>
```

```

<body>
  <h1>자전거(Bicycle)</h1>
  <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다. 바퀴
  <p class="example">최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국
  <p>대한민국에서는 1890년대 윤치호가 미국에서 돌아온 후에 처음 소개한 것으로 알려져 있으나 실제
</body>
</html>

```

절대 위치 `position:absolute`

페이지에 있는 다른 요소의 위치에 영향을 미치지 않고 지정한 위치로 간다.(원래 그 위치에 없는 듯이 동작한다.)

`position:absolute`

- 절대 위치로 지정된 요소와 기존 요소와 `width` 프로퍼티를 사용해 겹치지 않도록 배치한다.
- `<h1>` 요소 상단에는 기본적으로 마진이 생긴다. 이때 상단 마진을 없애기 위해 `margin` 프로퍼티를 사용한다.

```

<!DOCTYPE html>
<html>
  <head>
    <title>절대 위치</title>
    <style type="text/css">
      body {
        width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}
      h1 {
        position: absolute;
        top: 0px;
        left: 500px;
        width: 250px;}
      p {
        width: 450px;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다. 바퀴
    <p>최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기가 원
    <p>대한민국에서는 1890년대 윤치호가 미국에서 돌아온 후에 처음 소개한 것으로 알려져 있으나 실제
    <p>자전거는 타는 사람이 두 발로 발판을 밟아 체인으로 바퀴를 돌려 나아가게 만든 가볍고 편리한
    <p>자전거의 종류에는 산악용·도로용·어린이용 등이 있으며 그 밖에 두 명이 함께 탈 수 있는 탠덤

```

```
<p>자전거는 몸통을 이루는 뼈대, 방향을 조절하는 조향계, 앞·뒤 바퀴, 발판 등 구동장치, 변속장치
```

```
</body>
```

```
</html>
```

고정 위치 `position:fixed`

고정 위치는 절대 위치의 한 유형으로 **브라우저 창을 기준으로 요소의 위치를 지정한다**. 스크롤 하여도 동일한 위치에 머문다.

```
position:fixed
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>고정 위치</title>
    <style type="text/css">
      body {
        width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}
      h1 {
        position: fixed;
        top: 0px;
        left: 0px;
        padding: 10px;
        margin: 0px;
        width: 100%;
        background-color: #efefef;}
      p.example {
        margin-top: 100px;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <p class="example">자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는
    <p>최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기가 원
    <p>대한민국에서는 1890년대 윤치호가 미국에서 돌아온 후에 처음 소개한 것으로 알려져 있으나 실지
    <p>자전거는 타는 사람이 두 발로 발판을 밟아 체인으로 바퀴를 돌려 나아가게 만든 가볍고 편리한
    <p>자전거의 종류에는 산악용·도로용·어린이용 등이 있으며 그 밖에 두 명이 함께 탈 수 있는 탠덤
    <p>자전거는 몸통을 이루는 뼈대, 방향을 조절하는 조향계, 앞·뒤 바퀴, 발판 등 구동장치, 변속장치

  </body>
</html>
```

겹치는 요소 `z-index`

상대, 고정, 절대 위치를 사용할 경우 박스가 겹쳐질 수 있다. 이때 `z-index` 프로퍼티를 사용해 위에 위치할 요소를 제어한다. 값이 클 수록 위쪽에 나타난다.

`z-index`: [숫자]

```
<!DOCTYPE html>
<html>
  <head>
    <title>z-Index</title>
    <style type="text/css">
      body {
        width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}

      h1 {
        position: fixed;
        top: 0px;
        left: 0px;
        margin: 0px;
        padding: 10px;
        width: 100%;
        background-color: #efefef;
        z-index: 10;}

      p {
        position: relative;
        top: 70px;
        left: 70px;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다. 바퀴
    <p>최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기가 원
    <p>대한민국에서는 1890년대 윤치호가 미국에서 돌아온 후에 처음 소개한 것으로 알려져 있으나 실제
    <p>자전거는 타는 사람이 두 발로 발판을 밟아 체인으로 바퀴를 돌려 나아가게 만든 가볍고 편리한
    <p>자전거의 종류에는 산악용·도로용·어린이용 등이 있으며 그 밖에 두 명이 함께 탈 수 있는 탠덤
    <p>자전거는 몸통을 이루는 뼈대, 방향을 조절하는 조향계, 앞·뒤 바퀴, 발판 등 구동장치, 변속장치
  </body>
</html>
```

부유 요소 `float`

`float` 프로퍼티는 요소를 정상적인 흐름에 배치하여 부모 요소의 왼쪽 또는 오른쪽에 위치 한다. 부모 요소 안에 있는 모든 내용은 `float` 프로퍼티를 적용한 요소의 주위에 위치 한다.

`float` 프로퍼티를 사용할 경우에는 **width 프로퍼티를 함께 사용**한다. 그렇지 않으면 일관성이 떨어지거나 전체 너비로 커질 수 있다.

```

<title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
<body class="font-sans text-gray-800">
  <div class="container mx-auto px-4 py-8">
    <h1 class="text-2xl font-bold">
      (Bicycle)
    </h1>
    <p>
      "Life is like riding a bicycle. To keep your balance you must keep moving." - Albert Einstein
    </p>
    <p>
      自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다. 바퀴 두 개를 연결하는 구조
      자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기가 원조라고 주장하고 있는
      국에서는 1890년대 윤치호가 미국에서 돌아온 후에 처음 소개한 것으로 알려져 있으나 실제로 본인이 타고 다녔
    </p>
  </div>
</body>
</html>

```

float를 사용해 요소를 나란히 배치하는 방법

모든 요소를 부유 요소로 만들면 지정된 프로퍼티 값에 따라 차례대로 배치가 된다.

- 모두 `float:left`와 `width`값을 지정했을때
 - 단락이 왼쪽에서 오른쪽으로 차례대로 배치된다.
- 모두 `float:right`와 `width`값을 지정했을때
 - 단락이 오른쪽에서 왼쪽으로 차례대로 배치된다.

- **문제** : 브라우저 너비가 모자를 경우 특정 단락이 차례대로 안붙고 높이가 낮은 단락 아래에 붙는 문제가 발생한다. -_-
- **해결 방안** : `clear` 프로퍼티를 사용한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>float를 사용하여 요소를 나란히 배치하는 방법</title>
    <style type="text/css">
      body {
        width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}

      p {
        width: 230px;
        float: left;
        margin: 5px;
        padding: 5px;
        background-color: #efefef;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다.</p>
    <p>바퀴 두 개를 연결하는 구조 위에 안장을 두어 사람이 올라 탈 수 있도록 되어 있으며 발로 발걸
    <p>중국에서는 자행차(自行車)라고 부른다.</p>
    <p>최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기가 원
    <p>1839년에서 1840년 사이 스코틀랜드의 커크파트릭 맥밀란에 의해 페달식 크랭크가 발명되었다.
    <p>1861년 3월 프랑스의 에니스트 미슈가 처음으로 운전할 수 있게 하였다.</p>
  </body>
</html>
```

float 제거 `clear`

`clear` 프로퍼티는 부유 박스 뒤에 나타나는 박스가 인접하지 않도록 배치한다.

- 속성값
 - `left` : 박스의 왼쪽에 다른 요소가 위치하지 못하게 한다.
 - `right` : 박스의 오른쪽에 다른 요소가 위치하지 못하게 한다.
 - `both` : 박스의 양쪽에 다른 요소가 위치하지 못하게 한다.
 - `none` : 박스의 위치를 제약하지 않는다.

부유 요소의 부모: 문제점

부모 요소가 부유 요소만 포함다면 일부 브라우저는 부모 요소의 높이를 0픽셀로 처리한다.

부유 요소의 높이가 0이기에 위/아래 테두리 선이 합쳐져 화면 맨 위에 2픽셀 두께의 선으로 나타난다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>부유 요소의 부모 - 문제점</title>
    <style type="text/css">
      body {
        width: 752px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}
      div {
        border: 1px solid #665544;}
      p {
        width: 230px;
        float: left;
        margin: 10px;}
      .clear {
        clear: both;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <div>
      <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다.</p>
      <p>바퀴 두 개를 연결하는 구조 위에 안장을 두어 사람이 올라 탈 수 있도록 되어 있으며 발로 발걸<
      <p>중국에서는 자행차(自行車)라고 부른다.</p>
      <p class="clear">최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국<
      <p>1839년에서 1840년 사이 스코틀랜드의 커크파트릭 맥밀란에 의해 페달식 크랭크가 발명되었다.<
      <p>1861년 3월 프랑스의 에니스트 미슈가 처음으로 운전할 수 있게 하였다.</p>
    </div>
  </body>
</html>
```

부유 요소의 부모: 해결 방안

- 부유 요소 안에 별도의 요소를 추가한다.
- `overflow` 프로퍼티를 사용한다.
 - `overflow` 프로퍼티 값을 `auto`로 설정
 - `width` 프로퍼티 값을 100%로 설정


```

<!DOCTYPE html>
<html>
  <head>
    <title>부유 요소의 부모: 해결방안</title>
    <style type="text/css">
      body {
        width: 752px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}

      div {
        border: 1px solid #665544;
        overflow: auto;
        width: 100%;}

      p {
        width: 230px;
        float: left;
        margin: 10px;}

      .clear {
        clear: both;}
    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <div>
      <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다.</p>
      <p>바퀴 두 개를 연결하는 구조 위에 안장을 두어 사람이 올라 탈 수 있도록 되어 있으며 발로 발걸<
      <p>중국에서는 자행차(自行車)라고 부른다.</p>
      <p class="clear">최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국C
      <p>1839년에서 1840년 사이 스코틀랜드의 커크파트릭 맥밀란에 의해 페달식 크랭크가 발명되었다.<
      <p>1861년 3월 프랑스의 에니스트 미슈가 처음으로 운전할 수 있게 하였다.</p>
    </div>
  </body>
</html>

```

float를 사용한 다중 열 레이아웃 생성

`<div>` 요소를 사용해 각 열을 구분하고 `float` 프로퍼티를 사용해 열을 배치 한다.

- 설정 가이드
 - `width` : 열의 너비를 설정한다.
 - `float` : 서로의 옆에 열을 배치한다.
 - `margin` : 열 사이의 여백을 생성한다.

```

<!DOCTYPE html>

```

```
<html>
  <head>
    <title>2열 레이아웃</title>
    <style type="text/css">
      body {
        width: 960px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}

      .column1of2 {
        float: left;
        width: 620px;
        margin: 10px;}

      .column2of2 {
        float: left;
        width: 300px;
        margin: 10px;}

    </style>
  </head>
  <body>
    <h1>자전거(Bicycle)</h1>
    <div class="column1of2">
      <h3>자전거의 정의와 역사</h3>
      <p>자전거(自轉車)는 일반적으로 바퀴 두 개로 구성된 사람 힘으로 움직일 수 있는 기계이다.
      <p>최초의 자전거를 누가 만들었는지에 대한 의견은 분분하다. 독일, 프랑스, 영국이 모두 자기
      <h3>자전거의 원리와 종류</h3>
      <p>자전거는 타는 사람이 두 발로 발판을 밟아 체인으로 바퀴를 돌려 나아가게 만든 가볍고 편리
      <p>자전거의 종류에는 산악용·도로용·어린이용 등이 있으며 그 밖에 두 명이 함께 탈 수 있는
    </div>
    <div class="column2of2">
      <h3>함께 보기</h3>
      <ul>
        <li>자전거 구조</li>
        <li>자전거 타기</li>
        <li>자전거 길</li>
        <li>크리티컬 매스</li>
        <li>자전거 경기</li>
        <li>사이클</li>
      </ul>
    </div>
  </body>
</html>
```

화면 크기

사이트 방문자는 다양한 크기의 화면을 통해 사이트를 접속 한다.

다양한 크기의 화면에 최적화될 수 있도록 사이트를 디자인 해야 한다.

화면 해상도

해상도는 인치당 화면에 보이는 점의 수를 의미한다. 이 역시 다양한 기기에서 접속하기 때문에 다양한 해상도를 가진 기기에서 사이트를 접속한다.

페이지 크기

화면 크기와 해상도가 매우 다양하기 때문에 **페이지의 너비를 약 960~1000 픽셀로 만든다**(대부분의 사용자가 디자인을 볼 수 있다).

above the fold : 사용자가 스크롤 없이 볼 수 있는 페이지 영역(원래 신문을 반으로 접었을 때 상단에 있는 페이지 영역)

사용성 연구에 따르면 방문자는 아래로 두번 스크롤 한 다음에 페이지를 판단한다고 한다.

- 상단 570~600 픽셀 내에 중요한 정보를 배치
- 아래쪽에 더 많은 콘텐츠가 있다는 암시를 제공

고정 너비 레이아웃

사용자가 브라우저 창의 크기를 늘리거나 줄이더라도 크기가 변경되지 않는다. (픽셀 사용)

- 장점
 - 요소의 크기와 위치를 정확히 제어할 수 있다.
 - 항목의 모양이나 위치의 제어가 용이하다.
 - 화면 크기와 상관없이 텍스트 줄의 길이를 제어할 수 있다.
 - 이미지의 크기는 항상 동일하게 유지된다.
- 단점
 - 페이지의 가장자리에 큰 여백이 생길 수 있다.(큰 해상도 화면에서)
 - 사용자 해상도가 디자이너 화면보다 더 높을 경우 페이지와 글씨가 작아 보일 수 있다.
 - 사용자가 폰트 크기를 늘리면 텍스트가 할당된 영역에 맞지 않을 수 있다.
 - 유동 레이아웃보다 페이지가 세로로 길어 질 수 있다.

유동 레이아웃

사용자가 브라우저 창의 크기를 늘리거나 줄이면 커지거나 작아진다. (퍼센트/em 사용)

- 장점
 - 전체 브라우저 창을 채우기 위해 자동으로 확대되어, 페이지 주위에 여백이 생기지 않는다.
 - 페이지가 자동으로 축소 된다.
 - 디자이너의 의도 보다 폰트 크기를 크게 설정할 수 있다.
- 단점
 - 섹션 너비를 제어하지 않으면 의도치 않게 화면이 깨질 수 있다.

- 창이 넓으면 텍스트가 한줄에 표시되어 가독성이 떨어질 수 있다.
- 창이 좁으면 텍스트가 수많은 줄에 표시되어 가독성이 떨어질 수 있다.
- 창을 너무 작게 만들면 이미지가 텍스트 영역을 침범할 수 있다.

유동 레이아웃은 페이지의 일부분만 확대하거나 축소 시키도록 적용한다.

고정 너비 레이아웃 예제

고정 너비 레이아웃을 만들려면 페이지의 메인 박스의 너비와 높이를 픽셀로 설정한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>고정 너비 레이아웃</title>
    <style type="text/css">
      * {
        font-family: Arial, Verdana, sans-serif;
        color: #665544;
        text-align: center;}
      body {
        width: 960px;
        margin: 0 auto;}
      #content {
        overflow: auto;
        height: 100%;}
      #nav, #feature, #footer {
        background-color: #efefef;
        padding: 10px;
        margin: 10px;}
      .column1, .column2, .column3 {
        background-color: #efefef;
        width: 300px;
        float: left;
        margin: 10px;}
      li {
        display: inline;
        padding: 5px;}
    </style>
  </head>
  <body>
    <div id="header">
      <h1>로고</h1>
      <div id="nav">
        <ul>
          <li><a href="">홈</a></li>
          <li><a href="">상품</a></li>
          <li><a href="">서비스</a></li>
```

```
        <li><a href="">소개</a></li>
        <li><a href="">연락처</a></li>
    </ul>
</div>
</div>
<div id="content">
    <div id="feature">
        <p>특징</p>
    </div>
    <div class="article column1">
        <p>첫 번째 열</p>
    </div>
    <div class="article column2">
        <p>두 번째 열</p>
    </div>
    <div class="article column3">
        <p>세 번째 열</p>
    </div>
</div>
<div id="footer">
    <p>&copy; Copyright 2011</p>
</div>
</body>
</html>
```

유동 레이아웃 예제

디자인이 화면의 크기에 맞게 늘어날 수 있도록 퍼센트를 사용해 각 박스의 너비를 지정한다.

```
<!DOCTYPE html>
<html>
    <head>
        <title>유동 레이아웃</title>
        <style type="text/css">
            * {
                font-family: Arial, Verdana, sans-serif;
                color: #665544;
                text-align: center;}
            body {
                width: 90%;
                margin: 0 auto;}
            #content {
                overflow: auto;}
            #nav, #feature, #footer {
                margin: 1%;}
            .column1, .column2, .column3 {
```

```

        width: 31.3%;
        float: left;
        margin: 1%;}
.column3 {
    margin-right: 0%;}
li {
    display: inline;
    padding: 0.5em;}
#nav, #footer {
    background-color: #efefef;
    padding: 0.5em 0;}
#feature, .article {
    height: 10em;
    margin-bottom: 1em;
    background-color: #efefef;}
</style>
</head>
<body>
    <div id="header">
        <h1>로고</h1>
        <div id="nav">
            <ul>
                <li><a href="">홈</a></li>
                <li><a href="">상품</a></li>
                <li><a href="">서비스</a></li>
                <li><a href="">소개</a></li>
                <li><a href="">연락처</a></li>
            </ul>
        </div>
    </div>
    <div id="content">
        <div id="feature">
            <p>특징</p>
        </div>
        <div class="article column1">
            <p>첫 번째 열</p>
        </div>
        <div class="article column2">
            <p>두 번째 열</p>
        </div>
        <div class="article column3">
            <p>세 번째 열</p>
        </div>
    </div>
    <div id="footer">
        <p>&copy; Copyright 2011</p>
    </div>
</body>
</html>

```

레이아웃 그리드

- 구성 : 시각적 요소의 위치나 배열, 즉 페이지에 구성하는 방법을 의미
- 960픽셀 그리드
 - 보통 그리드를 이용해 한 페이지를 12개 단락으로 나누어 그 단락에 맞춰 디자인 한다.
 - 항목 간에 일관성 있는 비율과 영역 설정을 통해 전문적인 디자인이 가능하게 한다.
 - 장점
 - 다양한 디자인을 사용하는 여러 페이지 사이의 연속성을 만들 수 있다.
 - 다양한 페이지에서 사용자가 정보의 위치를 예측하는 데 도움을 준다.
 - 일관된 방식으로 사이트에 새로운 콘텐츠를 더 쉽게 추가할 수 있다.
 - 사이트의 디자인 작업을 일관된 방식으로 협업할 수 있다.
- 960픽셀 폭, 12열 그리드
 - 12개의 동일한 크기의 열이 있고 각 열의 너비는 60픽셀이다.
 - 각 열의 마진을 10픽셀로 설정한다.

CSS 프레임워크

레이아웃 그리드, 폼 스타일, 페이지의 인쇄용 버전등의 공통 작업을 더 쉽게 할 수 있는 코드를 제공하는데 목적이 있다.

- 장점
 - 동일한 작업의 대한 반복적인 코드 작성을 줄일 수 있다.
 - 다양한 브라우저 버전에서 이미 검증 됐다.
- 단점
 - 이미 정의된 클래스 이름을 사용해 스타일을 지정해야 한다.
 - 내 사이트에 불필요한 코드들을 포함할 수 있다.(코드 불로트 code bloat)
- 960.gs CSS
 - 그리드 레이아웃 프레임워크
 - <http://960.gs/>
 - 12열 그리드를 사용한 페이지 디자인의 템플릿을 제공
 - 사용방법
 - css 스크립트 임포트
 - 전체 페이지를 포함하는 최상위 부모 요소에 `class` 속성을 `container_12`로 설정. 그리고 부모 박스의 높이를 알 수 있도록 `clearfix`를 추가 설정
 - 각 열을 사용할 요소에 `class` 속성을 `grid_1` ~ `grid_12`로 설정

다중 스타일시트 @import, link

CSS 스타일 규칙을 모듈화해서 여러 파일로 만들 수 있다. 이때 다중 스타일시트를 추가하는 방법은 다음과 같다.

1. 스타일시트 페이지에 `@import` 규칙을 사용해 다른 스타일시트를 임포트한다.

```
@import url("tables.css");
@import url("typography.css");
body {
    color: #666666;
    background-color: #f8f8f8;
    text-align: center;}
#page {
    width: 600px;
    text-align: left;
    margin-left: auto;
    margin-right: auto;
    border: 1px solid #d6d6d6;
    padding: 20px;}
h3 {
    color: #547ca0;}
```

`@import` 규칙을 사용할 경우 다른 규칙 앞에 선언되어야 한다. (fifo 스크립트 읽기 방식 때문인듯)

1. HTML에 `<link>` 요소를 사용해 여러 스타일시트를 지정한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title> 다중 스타일시트 - 링크</title>
    <link rel="stylesheet" type="text/css" href="css/site.css" />
    <link rel="stylesheet" type="text/css" href="css/tables.css" />
    <link rel="stylesheet" type="text/css" href="css/typography.css" />
  </head>
  <body>
    <div id="page">
      <h1>센트럴 파크(Central Park) 자전거 대여</h1>
      <p class="intro">센트럴 파크 주변에서 탈 수 있는 자전거 대여:</p>
      <table>
        <tr class="head">
          <th></th>
          <th>시간당(Per hour)</th>
          <th>일당(Per day)</th></tr>
        <tr>
          <th>Cruiser</th>
          <td>$9</td>
          <td>$45</td>
        </tr>
        <tr>
```



```
<th>21 Speed</th>
<td>$15</td>
<td>$50</td>
</tr>
</table>
<h2>장소와 시간(Where and When)</h2>
<h3>로브 보트하우스(Loeb Boathouse)</h3>
<p>4월부터 11월까지 이용 가능하며 처음 방문 시 먼저 센트럴 파크에서 자전거 타기에 대한
<h2>예치금(Deposits)</h2>
<h3>현금 또는 신용카드(Cash or credit card)</h3>
<p>자전거 대여 시 $200의 예치금을 지불해야 한다.</p>
</div>
</body>
</html>
```

두 개의 규칙이 동일한 요소에 적용되면 문서에서 나중에 있는 규칙이 이전 규칙보다 우선한다.