JSP 기초 #2

2017-03-10

이승진

**학습 목표**

JSP 파일에서 for 루프의 구현 형태를 배운다

JSP 파일에서 if 문의 구현을 배운다.

**목차**

[1. for 루프(loop) 2](#_Toc476912227)

[1) loop1.jsp 2](#_Toc476912228)

[2) JSP에서 for 문장 구현 방법 4](#_Toc476912229)

[3) loop2.jsp 6](#_Toc476912230)

[4) loop3.jsp 7](#_Toc476912231)

[5) loop4.jsp 8](#_Toc476912232)

[2. if 문 10](#_Toc476912233)

[1) if1.jsp 10](#_Toc476912234)

[2) JSP Scriptet 태그에서 중괄호 필수 12](#_Toc476912235)

[3) if2.jsp 13](#_Toc476912236)

[4) if3.jsp 15](#_Toc476912237)

[3. 연습문제 16](#_Toc476912238)

[1) JSP 코드 분석 16](#_Toc476912239)

[2) 구현 실습 17](#_Toc476912240)

[4. 부록 18](#_Toc476912241)

# for 루프(loop)

JSP 파일에서 for 루프의 구현 형태를 살펴보자.

## loop1.jsp

### WebContent/jsp02폴더 생성

지난시간에 생성했던 lectur1 프로젝트에 WebContent/jsp02폴더를 생성한다.

생성방법:

이클립스의 화면 왼쪽 Project Explorer창에서 WebContent 폴더를 마우스 오른쪽 버튼으로 클릭하고

메뉴 - New - Folder 클릭

Folder name: 칸에 jsp02를 입력하고 Finish 버튼 클릭

### WebContent/jsp02/loop1.jsp 파일 생성.

생성방법:

이클립스의 화면 왼쪽 Project Explorer창에서 jsp02 폴더를 마우스 오른쪽 버튼으로 클릭하고

메뉴 - New - JSP File 클릭

File name: 칸에 loop1.jsp를 입력하고 Finish 버튼 클릭

WebContent/jsp02/loop1.jsp

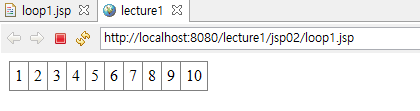
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <tr>  <%  for (int i=1; i <= 10; ++i) {  out.println(" <td>" + i + "</td>");  }  %>  </tr>  </table>  </body>  </html> |

|  |
| --- |
| <%  for (int i=1; i <= 10; ++i) {  out.println(" <td>" + i + "</td>");  }  %> |

for 문장을 하나의 JSP 스크립릿 태그로 묶었다.

out.println(...) 메소드 호출이 10번 반복되므로 이 메소드의 출력도 10번 반복될 것이다.

### loop1.jsp 실행



### loop1.jsp 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

<style>

table { border-collapse: collapse; }

td { padding: 5px; border: solid 1px gray; }

</style>

</head>

<body>

<table>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

<td>8</td>

<td>9</td>

<td>10</td>

</tr>

</table>

</body>

</html>

## JSP에서 for 문장 구현 방법

### 구현1

|  |
| --- |
| <%  for (int i=1; i <= 10; ++i) {  out.println(" <td>" + i + "</td>");  }  %> |

for 문장을 하나의 JSP 스크립릿 태그로 묶었다.

### 구현2

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <% out.println(" <td>" + i + "</td>"); %>  <% } %> |

나누어 묶어도 된다.

구현1과 구현2의 실행 결과는 동일하다.

### 구현3

|  |
| --- |
| <%  for (int i=1; i <= 10; ++i) {  out.println("i");  }  %> |

문자열 i를 10번 반복 출력한다.

### 구현4

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <% out.println("i"); %>  <% } %> |

문자열 i를 10번 반복 출력한다.

### 구현5

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <%= "i" %>  <% } %> |

문자열 i를 10번 반복 출력한다.

구현3, 구현4, 구현5의 실행 결과는 동일하다.

### 구현6

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  i  <% } %> |

위 코드에서 i 부분은 JSP 태그 외부이다.

JSP 태그 외부 부분은, 있는 그대로 출력된다.

JSP 태그 외부의 i는 Java 변수가 아니고 평범함 문자열이다.

위 코드는 문자열 i를 10번 반복 출력한다.

구현3, 구현4, 구현5, 구현6의 실행 결과는 동일하다.

### 구현7

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <%= i %>  <% } %> |

위 코드에서 i 부분은 JSP 표현식 태그 내부이다.

따라서 Java 변수 i의 값이 반복해서 출력된다.

위 코드는 아래와 같이 출력된다.

|  |
| --- |
| 1  2  3  4  5  6  7  8  9  10 |

### 구현8

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <td><%= i %></td>  <% } %> |

위 코드에서 i 부분은 JSP 표현식 태그 내부이다.

위 코드에서 <td>와 </td> 부분은 JSP 태그 외부이다.

JSP 태그 외부 부분은, 있는 그대로 출력된다.

따라서 위 코드는 아래와 같이 출력된다.

|  |
| --- |
| <td>1</td>  <td>2</td>  <td>3</td>  <td>4</td>  <td>5</td>  <td>6</td>  <td>7</td>  <td>8</td>  <td>9</td>  <td>10</td> |

### 구현9

|  |
| --- |
| <% for (int i=1; i <= 10; ++i) { %>  <td>i</td>  <% } %> |

위 코드에서 <td>i</td> 부분은 JSP 태그 외부이다. i는 Java 변수가 아니다.

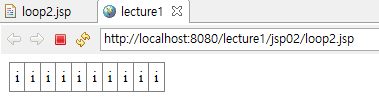
따라서 위 코드는 아래와 같이 출력된다.

|  |
| --- |
| <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td>  <td>i</td> |

## loop2.jsp

### WebContent/jsp02/loop2.jsp 파일 생성.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <tr>  <% for (int i=1; i <= 10; ++i) { %>  <td>i</td>  <% } %>  </tr>  </table>  </body>  </html> |



### 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

<style>

table { border-collapse: collapse; }

td { padding: 5px; border: solid 1px gray; }

</style>

</head>

<body>

<table>

<tr>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

<td>i</td>

</tr>

</table>

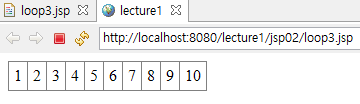
</body>

</html>

## loop3.jsp

### WebContent/jsp02/loop3.jsp 파일 생성.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <tr>  <% for (int i=1; i <= 10; ++i) { %>  <td><%= i %></td>  <% } %>  </tr>  </table>  </body>  </html> |



### 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

<style>

table { border-collapse: collapse; }

td { padding: 5px; border: solid 1px gray; }

</style>

</head>

<body>

<table>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

<td>8</td>

<td>9</td>

<td>10</td>

</tr>

</table>

</body>

</html>

## loop4.jsp

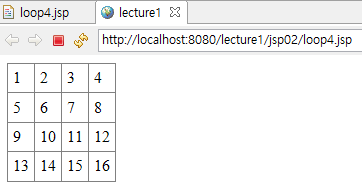
### WebContent/jsp02/loop4.jsp 파일 생성.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <% for (int i=0; i <= 3; ++i) { %>  <tr>  <% for (int j=1; j <= 4; ++j) { %>  <td><%= i \* 4 + j %></td>  <% } %>  </tr>  <% } %>  </table>  </body>  </html> |

|  |
| --- |
| <% for (int i=0; i <= 3; ++i) { %>  <tr>  <% for (int j=1; j <= 4; ++j) { %>  <td><%= i \* 4 + j %></td>  <% } %>  </tr>  <% } %> |

위 코드는 아래의 코드와 동일하다.

|  |
| --- |
| <%  for (int i=0; i <= 3; ++i) {  out.println("<tr>");  for (int j=1; j <= 4; ++j) {  out.print("<td>");  out.print( i \* 4 + j )  out.println("</td>");  }  out.println("</tr>");  }  %> |



### 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

<style>

table { border-collapse: collapse; }

td { padding: 5px; border: solid 1px gray; }

</style>

</head>

<body>

<table>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

</tr>

<tr>

<td>5</td>

<td>6</td>

<td>7</td>

<td>8</td>

</tr>

<tr>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

</tr>

<tr>

<td>13</td>

<td>14</td>

<td>15</td>

<td>16</td>

</tr>

</table>

</body>

</html>

# if 문

## if1.jsp

### WebContent/jsp02/if1.jsp 파일 생성.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.\*" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  </head>  <body>  <%  Date now = new Date();  Calendar calendar = GregorianCalendar.getInstance();  calendar.setTime(now);  int year = calendar.get(Calendar.YEAR);  int month = calendar.get(Calendar.MONTH) + 1;  int day = calendar.get(Calendar.DAY\_OF\_MONTH);  int hour = calendar.get(Calendar.HOUR\_OF\_DAY);  int minute = calendar.get(Calendar.MINUTE);  int second = calendar.get(Calendar.SECOND);  %>  <% if (hour < 12) { %>  Good Morning!  <% } else if (hour < 18){ %>  Good Afternoon!  <% } else { %>  Good Evening!  <% } %>  <br />  지금은 <%= year %>년 <%= month %>월 <%= day %>일 <%= hour %>시 <%= minute %>분 <%= second %> 초 입니다.  </body>  </html> |

|  |
| --- |
| Date now = new Date();  Calendar calendar = GregorianCalendar.getInstance();  calendar.setTime(now);  int year = calendar.get(Calendar.YEAR);  int month = calendar.get(Calendar.MONTH) + 1;  int day = calendar.get(Calendar.DAY\_OF\_MONTH);  int hour = calendar.get(Calendar.HOUR\_OF\_DAY);  int minute = calendar.get(Calendar.MINUTE);  int second = calendar.get(Calendar.SECOND); |

현재 시각의 연, 월, 일, 시, 분 초를 구하는 Java 코드이다.

### 참고:

|  |
| --- |
| int month = calendar.get(Calendar.MONTH) + 1; |

calendar.get(Calendar.MONTH) 메소드 리턴 값의 범위가 0~11 이기 때문에, +1 해주었다.

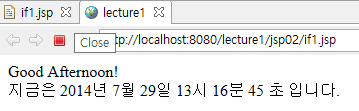
|  |
| --- |
| <% if (hour < 12) { %>  Good Morning!  <% } else if (hour < 18){ %>  Good Afternoon!  <% } else { %>  Good Evening!  <% } %> |

JSP 태그의 외부인 Good Morning!, Good Afternoon!, Good Evening! 부분은 있는 그대로 출력된다.

hour < 12 조건식이 true이면 Good Morning! 부분만 출력되고,

hour < 18 조건식이 true이면 Good Afternoon! 부분만 출력되고,

그렇지 않으면 Good Evening! 부분만 출력된다.



### if1.jsp 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

</head>

<body>

Good Evening!

<br />

지금은 2014년 7월 22일 20시 11분 33 초 입니다.

</body>

</html>

## JSP Scriptet 태그에서 중괄호 필수

<% if (hour < 12) %>

Good Morning!

<% else if (hour < 18) %>

Good Afternoon!

<% else %>

Good Evening!

위와 같이 중괄호를 생략하는 것은 위험하다.

아래와 같이 중괄호로 묶어주어야 한다.

<% if (hour < 12) **{** %>

Good Morning!

<% **}** else if (hour < 18) **{** %>

Good Afternoon!

<% **}** else **{** %>

Good Evening!

<% **}** %>

<% for (int i=1; i <= 10; ++i) %>

<td><%= i %></td>

위와 같이 중괄호를 생략하는 것은 위험하다.

아래와 같이 중괄호로 묶어주아야 한다.

<% for (int i=1; i <= 10; ++i) **{** %>

<td>i</td>

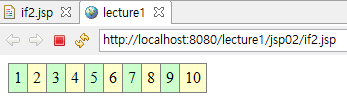
<% **}** %>

## if2.jsp

### WebContent/jsp02/if2.jsp 파일 생성.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <tr>  <% for (int i=1; i <= 10; ++i) { %>  <% if (i % 2 == 0) { %>  <td style="background-color: #ffffcc"><%= i %></td>  <% } else { %>  <td style="background-color: #ccffcc"><%= i %></td>  <% } %>  <% } %>  </tr>  </table>  </body>  </html> |

#ffffcc는 밝은 노란색이고, #ccffcc는 밝은 녹색이다.



### 실행결과 소스보기

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>lecture1</title>

<style>

table { border-collapse: collapse; }

td { padding: 5px; border: solid 1px gray; }

</style>

</head>

<body>

<table>

<tr>

<td style="background-color: #ccffcc">1</td>

<td style="background-color: #ffffcc">2</td>

<td style="background-color: #ccffcc">3</td>

<td style="background-color: #ffffcc">4</td>

<td style="background-color: #ccffcc">5</td>

<td style="background-color: #ffffcc">6</td>

<td style="background-color: #ccffcc">7</td>

<td style="background-color: #ffffcc">8</td>

<td style="background-color: #ccffcc">9</td>

<td style="background-color: #ffffcc">10</td>

</tr>

</table>

</body>

</html>

### 구현 실습

if2.jsp에서 if 문 대신 ? : 연산자를 사용하면, 훨씬 더 간단하게 구현할 수 있다.

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  table { border-collapse: collapse; }  td { padding: 5px; border: solid 1px gray; }  </style>  </head>  <body>  <table>  <tr>  <% for (int i=1; i <= 10; ++i) { %>  <td style="background-color: <%= %>" ><%= i %></td>  <% } %>  </tr>  </table>  </body>  </html> |

<%= %> 태그 내부에 적당한 Java 표현식을 추가하라.

? : 연산자를 사용해서 표현식을 구현하라.

i 값이 짝수일 때는 "#ffffcc" 값이고, 그렇지 않을 때는 "#ccffcc" 값인 표현식이어야 한다.

## if3.jsp

### WebContent/jsp02/if3.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.\*" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  </head>  <body>  <%  Date now = new Date();  Calendar calendar = GregorianCalendar.getInstance();  calendar.setTime(now);  int second = calendar.get(Calendar.SECOND);  %>  지금은 <%= second %> 초 입니다.  <% if (second % 2 == 0) return; %>  <%= second %>는 홀수입니다.  </body>  </html> |

|  |
| --- |
| <% if (second % 2 == 0) return; %> |

JSP 파일 실행 중간에 return 하면,

return 문 아래 부분은 실행되지 않으므로, 웹브라우저에 출력되지도 않는다.

second 값이 짝수일 때 위 if 문이 true가 되어 return 하게 된다.

따라서 second 값이 홀수일때에만 아래 코드가 실행될 수 있다.

|  |
| --- |
| <%= second %>는 홀수입니다. |

# 연습문제

## JSP 코드 분석

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<% for (int i=1; i <= 10; ++i) { %>

4 \* i = 4 \* i

<% } %>

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<% for (int i=1; i <= 10; ++i) { %>

<%= 4 \* i %> = <%= 4 \* i %>

<% } %>

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<% for (int i=1; i <= 10; ++i) { %>

<%= 4 %> \* <%= i %> = <%= 4 \* i %>

<% } %>

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<% for (int i=1; i <= 10; ++i) { %>

4 \* <%= i %> = <%= 4 \* i %>

<% } %>

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<tr>

<% for (int i=1; i <= 10; ++i) { %>

<td>4 \* <%= i %> = <%= 4 \* i %></td>

<% } %>

</tr>

아래 코드의 실행 결과 출력되는 내용을 적으시오.

<td>

<% for (int i=1; i <= 10; ++i) { %>

4 \* <%= i %> = <%= 4 \* i %><br />

<% } %>

</td>

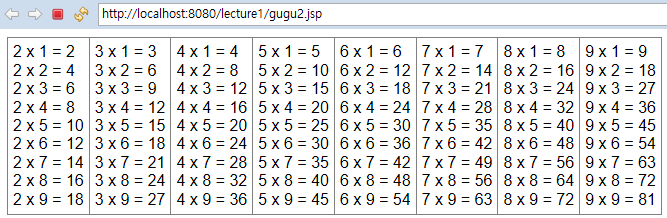
위 코드들의 실행 결과 출력을 계산해서 적어보라.

그리고 나서, 위 코드들을 실행해 보고 실행 결과 출력과 자신이 계사한 것을 비교해 보자.

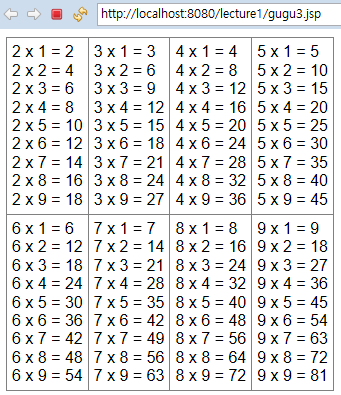
## 구현 실습

다음과 같이 출력하는 gugu2.jsp를 구현하라.

최대한 간결하게 구현하라.



다음과 같이 출력하는 gugu3.jsp를 구현하라.



# 부록

### WebContent/jsp02/loop5.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>lecture1</title>  <style>  div.outter { height: 25px; }  div.inner { display: inline-block; width:2px; height:25px; }  </style>  </head>  <body>  <%  for (int red=0; red <= 255; red += 15) {  out.println("<div class='outter'>");  for (int green=0; green <= 255; green += 15) {  for (int blue=0; blue <= 255; blue += 15) {  String color = String.format("#%02x%02x%02x", red, green, blue);  String s = String.format("<div class='inner' style='background-color:%s'></div>", color);  out.print(s);  }  }  out.println("</div>");  }  %>  </body>  </html> |

(줄8) <div class='outter'> 태그들에 서식을 지정한다. 이 태그들의 높이를 20 픽셀로 지정한다.

(줄9) <div class='inner'> 태그들에 서식을 지정한다.

div 태그는 웹브라우저에서 그려질 때 기본으로 가로 한줄을 혼자 다 차지하는 형태로 그려진다.

즉 div 태그 뒤에 줄이 바뀌어서, 뒷 태그들은 아래 줄에 그려진다.

그런데 display: inline-block 으로 서식을 지정하면 div 태그 뒤에 줄이 바꾸지 않고,

뒷 태그들은 오른쪽에 계속 그려진다.

(줄19) String 클래스의 format 메소드는 C 언어의 printf 함수와 유사하다.

이 메소드는 파라미터로 주어진 값을 바탕으로 새 문자열을 생성하여 리턴한다.

이 메소드의 첫째 파라미터는 C 언어의 printf 함수의 첫째 파라미터와 같은 역할을 한다.

C 언어의 printf 사용법을 설명한 책에서 포맷 스트링(format string)에 대한 설명을 찾아서 공부하라.

포맷 스트링에서 "%02x"의 의미는 뒤에 오는 정수 파라미터를 두 자리의 16진수로 출력하라는 것.

"%02x%02x%02x"의 의미는 뒤에 오는 3개의 정수 파라미터를 각각 두 자리의 16진수로 총 6자리의 16진수로 출력하라는 것.

포맷 스트링에서 "#" 문자는 아무 의미가 없어서 그냥 그 문자 그대로 출력된다.

예를들어 변수 red 값이 255이고, 변수 green 값이 128이고, 변수 blue 값이 160이라면,

이들은 각각 16진수로 FF, 80, A0 이므로,

String.format("#%02x%02x%02x", red, green, blue) 메소드는 "#FF80A0" 문자열을 생성하여 리턴한다.

HTML에서 색을 지정할 때는 "#FF80A0" 형태의 16진수 문자열을 사용한다.

"#FF80A0" 색은 빛의 삼요소인 red를 255만큼, green을 128만큼, blue를 160만큼 섞은 색이다.

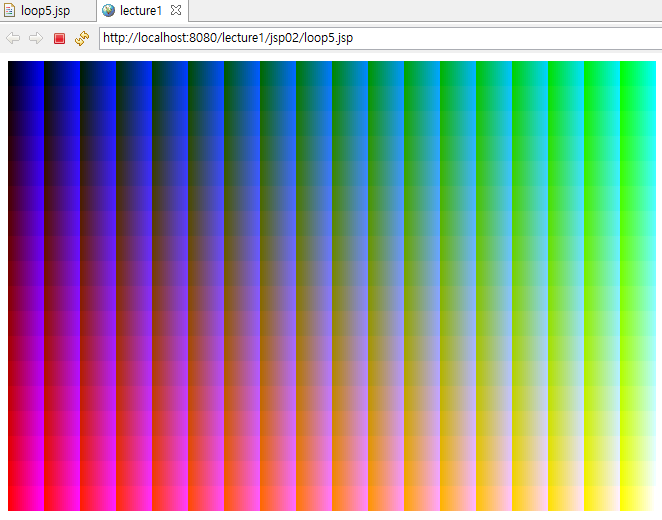
색을 지정할 때 red, green, blue 값의 범위는 최소 0 부터 최대 255 이다.

10진수 255는 16진수 FF 이므로 색의 값의 범위는 16진수로 최소 00 부터 최대 FF 이다.

#000000 -> 검정색 #FFFFFF -> 흰색 #FF0000 -> 빨간색 #808080 -> 회색

#00FF00 -> 녹색 #0000FF -> 파란색 #FFFF00 -> 노란색 #FF00FF -> 보라색

### 실행결과



위 실행결과에서 <div class='outter'> 태그는 가로로 한줄을 차지하고 있다.

<div class='outter'> 태그는 255 / 15 = 17번 반복해서 출력된다.

즉 위 실행결과는 17행의 <div class='outter'> 태그로 구성된다.

<div class='outter'> 태그의 높이는 25px 이다.

25px \* 17행 = 426px 이므로 위 실행결과에서 색이 칠해진 영역의 높이는 426 픽셀이다.

<div class='outter'> 태그 내부에는

여러개의 <div class='inner' style='background-color:......'></div> 태그로 채워져 있다.

for 루프 하나가 255 / 15 = 17번 반복하므로,

하나의 <div class='outter'> 태그 내부에는 17 \* 17 = 289 개의

<div class='inner' style='background-color:......'></div> 태그가 들어있다.

<div class='inner' style='background-color:......'></div> 태그들은 display: inline-block 서식이므로

줄바꿈 없이 서로 연이어 붙어서 그려진다.

<div class='inner' style='background-color:......'></div> 태그의 가로 폭은 2px 이다.

289개 \* 2px = 578px 이므로 위 실행결과에서 색이 칠해진 영역의 가로 폭은 578 픽셀이다.