

1장 프로그램의 발전 과정

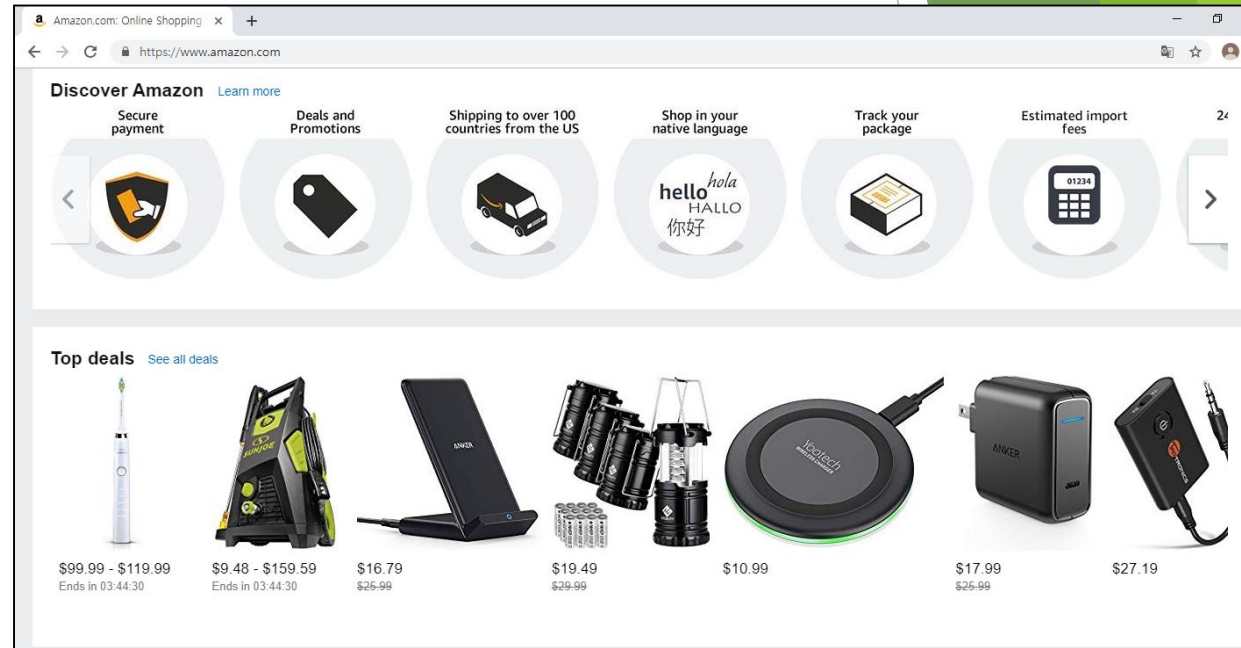
1.1 클라이언트 PC 기반 프로그램

1.2 클라이언트-서버 기반 프로그램 동작 방식

1.3 웹 기반 프로그램 동작 방식

1.1 클라이언트 PC 기반 프로그램

- 프로그램 발전 과정



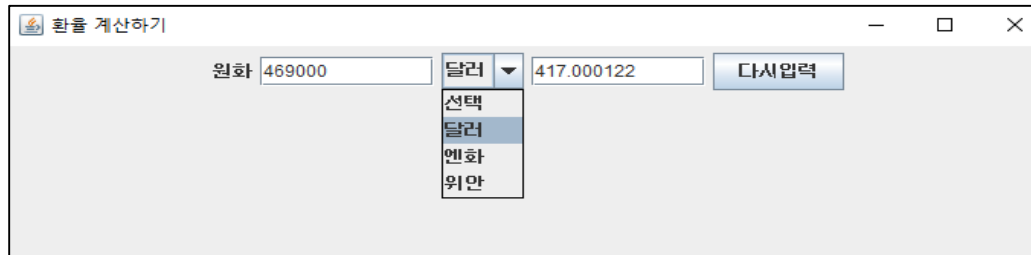
1. 클라이언트 PC 기반 프로그램
2. 클라이언트/서버 프로그램
3. 웹 기반 프로그램

1.1 클라이언트 PC 기반 프로그램

- 클라이언트 PC 기반 프로그램의 특징

➤ 클라이언트 PC에서 실행하면서 모든 기능을 수행

- 클라이언트 PC에서 실행되는 자바 로 구현한 환율 계산기

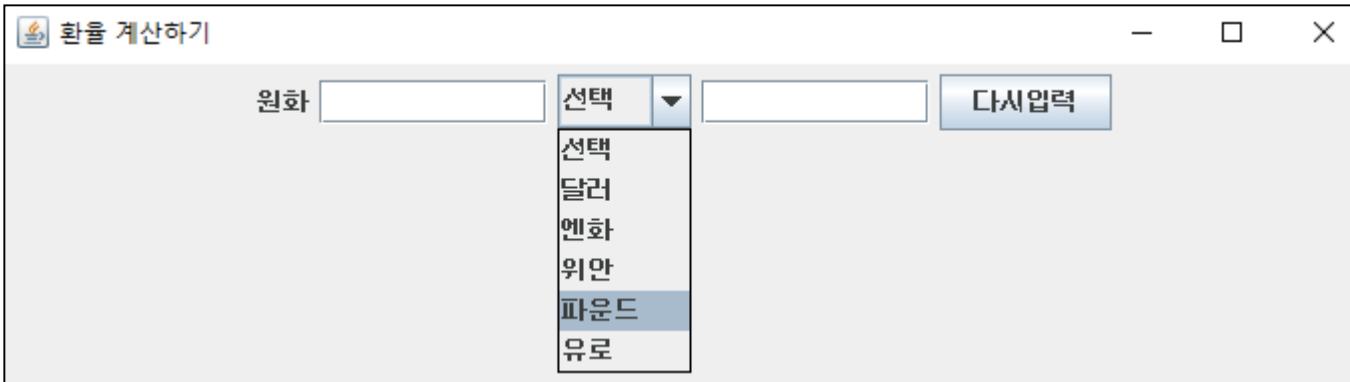


코드 1-1 클라이언트 환율 계산기 프로그램

```
private void calculate() {  
    float won = Float.parseFloat(operand1.getText());  
    String result = null;  
    String operator = opSelection.getSelectedItem().toString();  
  
    if (operator.equals("달러")) {  
        result = String.format("%.6f", won / USD_RATE);  
    }  
    else if (operator.equals("엔화")) {  
        result = String.format("%.6f", won / JPY_RATE);  
    }  
    else if (operator.equals("위안")) {  
        result = String.format("%.6f", won / CNY_RATE);  
    }  
    txtResult.setText(result);  
}  
...
```

1.1 클라이언트 PC 기반 프로그램

- 파운드와 유로화 변환 기능이 추가된 환율 계산기



코드 1-2 기능이 변경된 환율 계산기 프로그램(pro01/src/ex01/Calculator.java)

```
private void calculate() {  
    ...  
    else if (operator.equals("파운드")) {  
        result = String.format("%.6f", won / GBP_RATE);  
    }  
    else if (operator.equals("유로")) {  
        result = String.format("%.6f", won / EUR_RATE);  
    }  
    txtResult.setText(result);  
}
```

선택 박스에서 '파운드' 또는
'유로' 선택 시 원화를 파운드 또
는 유로로 환산하는 코드 추가



소스 코드를 추가 후 다시 모든
클라이언트 PC에 설치하거나
업그레이드를 해야 합니다.

1.1 클라이언트 PC 기반 프로그램

- 클라이언트 PC 기반 프로그램의 문제점

1. 프로그램이 변경될 때마다 일일이 다시 설치해야함
2. 데이터베이스 접속 정보와 같은 정보가 쉽게 노출 될 수 있어 보안에 취약함

1.2 클라이언트-서버 기반 프로그램 동작 방식

- 클라이언트- 서버 기반 프로그램 구조



- 클라이언트- 서버 기반 프로그램 특징

- 클라이언트 프로그램은 화면 기능과 데이터 송·수신만 제공
- 모든 기능은 서버에서 수행

1.2 클라이언트-서버 기반 프로그램 동작 방식

- 서버가 담당하는 계산기 기능

코드 1-3 환율 계산기 서버의 기능(pro01/src/ex02/RateServer.java)

```
public class RateServer {
    public static void main(String[] args) {
        ...
        try {
            serverSocket = new ServerSocket(5434);
            System.out.println("서버 실행 중... ");

            while(true) {
                s1 = serverSocket.accept();
                is = s1.getInputStream();
                os = s1.getOutputStream();
                br = new BufferedReader(new InputStreamReader(is));
                String data = br.readLine();
                System.out.println("서버 수신 데이터:"+data);
                String result = calculate(data);
                System.out.println(result);

                bw = new BufferedWriter(new OutputStreamWriter(os));
                pw = new PrintWriter(bw,true);
                pw.println(result);
                pw.close();
            }

        } catch(IOException ie) {
            ie.printStackTrace();
        }
        ...
    }
}
```

1.2 클라이언트-서버 기반 프로그램 동작 방식

- 클라이언트가 담당하는 기능

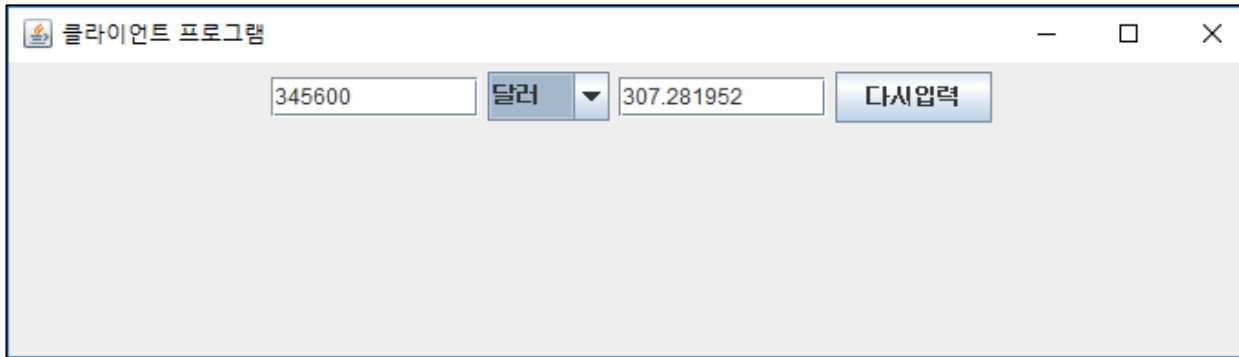
코드 1-4 환율 계산기 클라이언트의 기능(pro01/src/ex02/RateClient.java)

```
...
private void calculate() {
    float won = Integer.parseInt(operand1.getText());
    String result = null;
    String operator = opSelection.getSelectedItem().toString();
    InputStream is;
    BufferedReader br;
    BufferedWriter bw;
    OutputStream os;
    PrintWriter pw = null;
    try {
        Socket s1 = new Socket("127.0.0.1", 5434);
        os = s1.getOutputStream();
        is = s1.getInputStream();
        System.out.println("전송 데이터: "+won+", "+operator);
        bw = new BufferedWriter(new OutputStreamWriter(os));
        pw = new PrintWriter(bw, true);
        pw.println(won+", "+operator);

        br = new BufferedReader(new InputStreamReader(is));
        result = br.readLine();
        System.out.println("클라이언트 수신 데이터: "+result);
        txtResult.setText(result);
        s1.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
...
```


1.2 클라이언트-서버 기반 프로그램 동작 방식

- 실행 결과



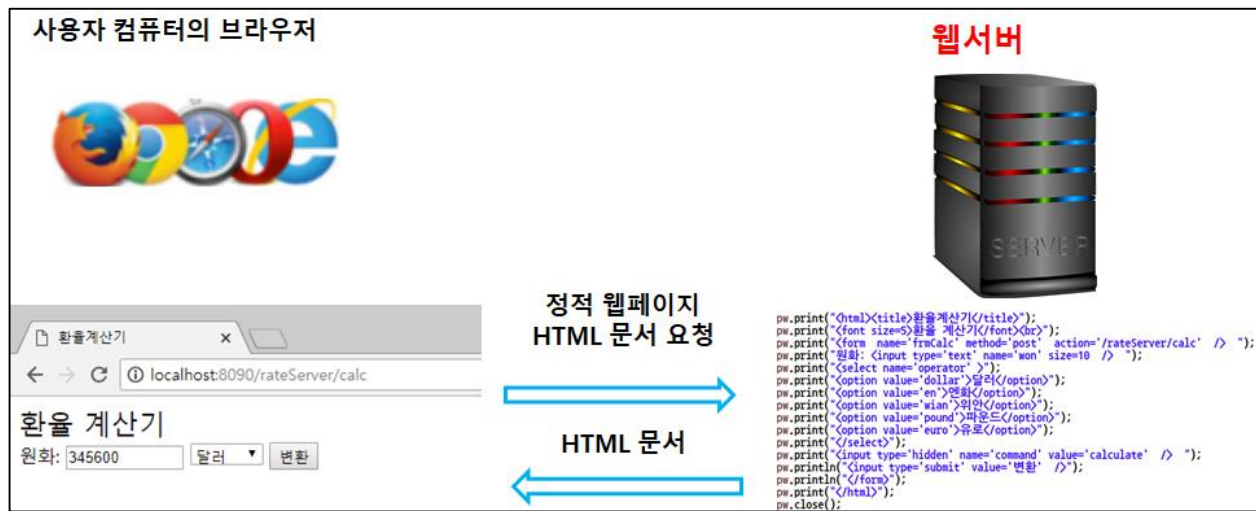
- 클라이언트-서버 기반 프로그램의 특징

- 기능(로직)이 변경되어도 모두 서버에서 처리하므로 클라이언트 프로그램은 수정할 필요가 없음
- 중요한 기능은 서버에서 처리하므로 보안 측면에서도 우수함
- 그러나 사용자에게 관련된 화면 기능이 바뀌면 클라이언트 프로그램도 수정해서 다시 설치해야함

1.3 웹 기반 프로그램 동작 방식

- 웹 기반 프로그램 구조

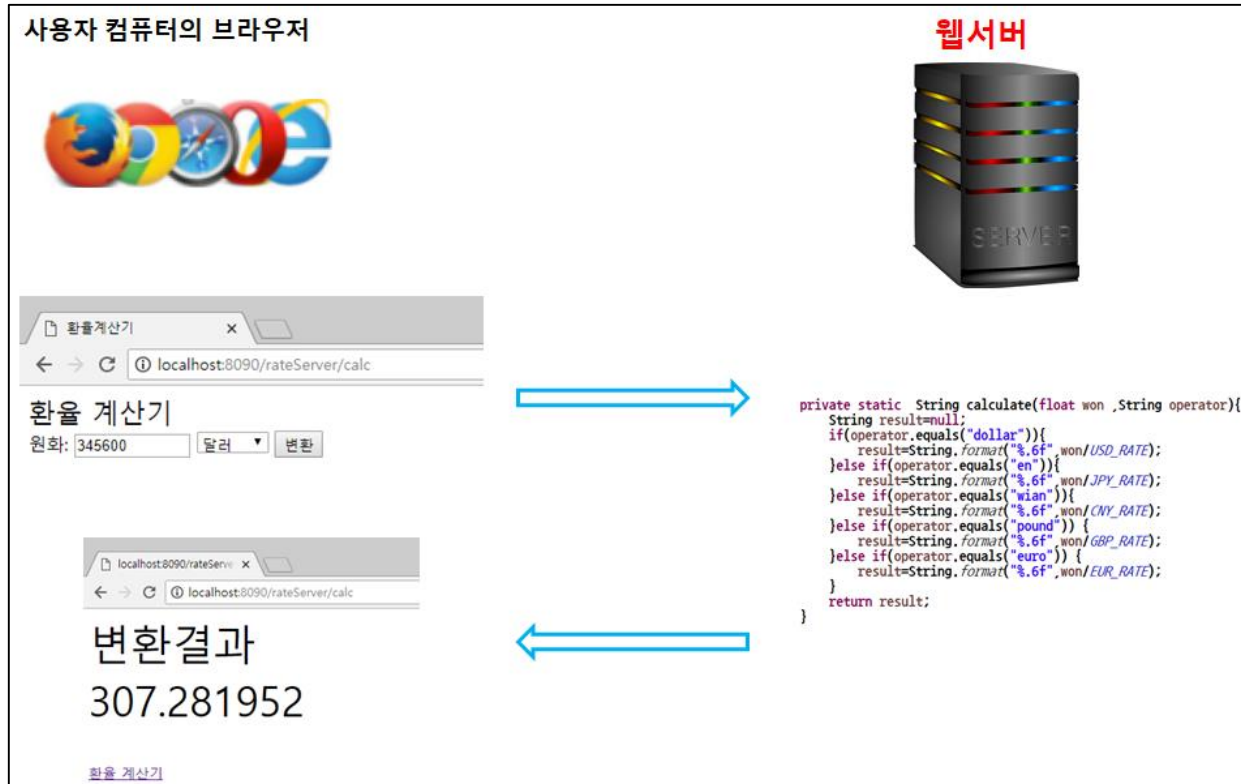
➤ 화면과 데이터 처리를 모두 서버에서 수행함



- ① 브라우저에서 웹 페이지를 요청합니다.
- ② 브라우저는 서버에서 전송된 HTML을 화면에 표시합니다.

1.3 웹 기반 프로그램 동작 방식

- 웹 기반 환율 계산기 동작 과정



- ① 화면에서 처리할 데이터를 입력 후 서버에 요청합니다.
- ② 서버는 브라우저에서 전송된 데이터를 받아서 처리합니다.
- ③ 서버는 처리 결과를 브라우저로 전송해서 결과를 보여줍니다.

1.3 웹 기반 프로그램 동작 방식

- 웹 기반 프로그램의 특징

- 화면과 로직을 서버에서 모두 처리하므로 클라이언트가 특별히 수행할 작업이 없음
- 모든 기능이 서버에서 처리되므로 보안 면에서도 월등히 우수함

1.3 웹 기반 프로그램 동작 방식

- 화면과 계산 기능을 모두 처리하는 환율 계산기

코드 1-5 http 환율 계산 웹 서버(rateServer/src/ex01/RateServlet.java)

```
...  
if(command!= null && command.equals("calculate")){  
    String result = calculate(Float.parseFloat(won),operator);  
    pw.print("<html><font size=10>변환결과</font><br>");  
    pw.print("<html><font size=10>"+result+"</font><br><br><br>");  
    pw.print("<a href='/rateServer/calc'>환율 계산기 </a>");  
    return;  
}
```

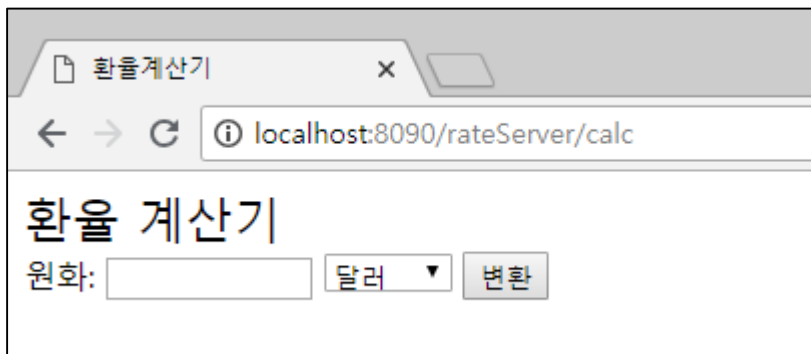
변환한 결과를
표시합니다.

계산기 화면을
표시합니다.

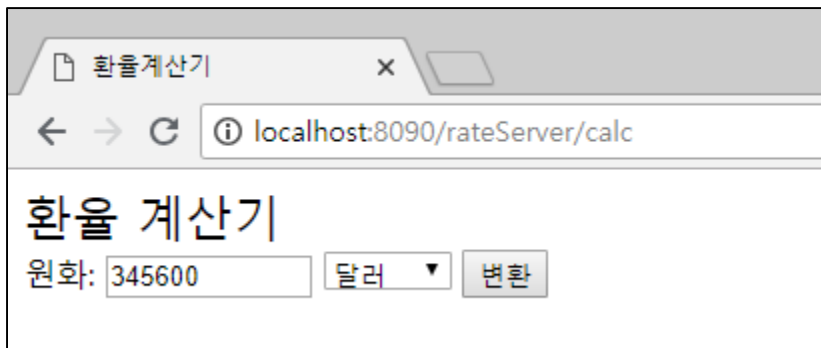
```
pw.print("<html><title>환율 계산기</title>");  
pw.print("<font size=5>환율 계산기</font><br>");  
pw.print("<form name='frmCalc' method='post' action='/rateServer/calc' /> ");  
pw.print("원화: <input type='text' name='won' size=10 /> ");  
pw.print("<select name='operator' >");  
pw.print("<option value='dollar'>달러</option>");  
pw.print("<option value='en'>엔화</option>");  
pw.print("<option value='wian'>위안</option>");  
pw.print("<option value='pound'>파운드</option>");  
pw.print("<option value='euro'>유로</option>");  
pw.print("</select>");  
pw.print("<input type='hidden' name='command' value='calculate' /> ");  
pw.println("<input type='submit' value='변환' />");  
pw.println("</form>");  
pw.print("</html>");  
pw.close();  
...
```

1.3 웹 기반 프로그램 동작 방식

- 브라우저에서 <http://localhost:8090/rateServer/calc>로 요청 시 계산기 화면 표시



- 텍스트 필드에 변환할 원화를 입력 후 변환 버튼 클릭



1.3 웹 기반 프로그램 동작 방식

- 서버에서 전달된 원화를 달러로 변환 후 브라우저로 결과 출력

