

Chương 11

Lập trình Web chạy ở client bằng Java

11.0 Dẫn nhập

11.1 Tổng quát về lập trình Web

11.2 Tổng quát về lập trình Web chạy ở phía client

11.3 Tổng quát về Java Applet

11.4 Qui trình điển hình để xây dựng và dùng Applet

11.5 Viết và dùng java script

11.6 Thí dụ viết trang Web có dùng java script

11.7 Kết chương



11.0 Dẫn nhập

- ❑ Chương này sẽ trình bày kiến thức về lập trình Web chạy ở phía client dùng ngôn ngữ Java và Javascript.
- ❑ Chương này sẽ giới thiệu công dụng của Applet, cấu trúc tổ chức của Applet, cách viết, kiểm thử 1 applet trước khi đóng gói nó vào file *.jar để phân phối cho website sử dụng nó.
- ❑ Chương này cũng giới thiệu cách viết trang Web có sử dụng Applet, cách viết tag <applet> để đặc tả thông tin về applet sử dụng.
- ❑ Chương này cũng trình bày cách tính chất của Javascript, công dụng của nó và giới thiệu 1 thí dụ về cách viết trang web có dùng các đoạn lệnh Javascript.



11.1 Tổng quát về lập trình Web

- ❑ Hiện nay, mạng Internet đã là mạng toàn cầu, nó nối kết mọi máy tính trên thế giới lại để chúng có thể giao tiếp lẫn nhau.
- ❑ Có nhiều phần mềm mạng đang được dùng trên mạng Internet, trong đó có thể nói hệ thống World Wide Web (WWW) là phần mềm mạng được sử dụng phổ biến nhất.
- ❑ Hệ thống WWW gồm nhiều module phần mềm mạng chạy trên nhiều máy, chúng thuộc 1 trong 2 loại : Web server chuyên cung cấp các trang Web, Web client chuyên yêu cầu download các trang Web.
- ❑ Đơn vị trao đổi cơ bản trong hệ thống WWW là trang Web. Trang Web là 1 file văn bản thô được viết theo ngôn ngữ đánh dấu siêu văn bản HTML (HyperText Markup Language).



11.1 Tổng quát về lập trình Web

- ❑ Lúc đầu, ngôn ngữ HTML chỉ định nghĩa các lệnh (tag) để định nghĩa nội dung có định dạng (giống như ngôn ngữ RTF miêu tả tài liệu Word). Tag lệnh thiết yếu nhất của HTML là ``, nó cho phép ta liên kết trang Web này với trang Web khác (hay tổng quát hơn là 1 tài nguyên Internet bất kỳ).
- ❑ Các tag HTML cơ bản chưa thể biến trang Web thành phần mềm tương tác trực quan với người dùng như các ứng dụng truyền thống trên Windows \Rightarrow cần 1 số tag lệnh mới để biến trang Web thành ứng dụng.
- ❑ Lưu ý rằng lúc đầu, trang Web nằm ở Web server, nhưng khi cần nó sẽ được download về phía client \Rightarrow thành phần phần mềm được nhúng vào trang Web có thể được xử lý trước ở Server hay sau khi đã được download về phía client \Rightarrow có 2 công nghệ lập trình Web khác nhau :



11.1 Tổng quát về lập trình Web

1. lập trình Web chạy ở client dùng các lệnh sau :

- `<object> </object>` : nhúng 1 phần tử COM, ActiveX Control,...
- `<applet code=URL ...>` : nhúng 1 Java Applet
- `<script> </script>` : tạo 1 đoạn chương trình (thường là n hàm xử lý)

2. lập trình Web chạy ở server dùng các lệnh sau :

- `<script runat=server> </script>` : tạo 1 đoạn chương trình (thường là n hàm xử lý)
- `<form action=URL> </form>` : tạo 1 form giao diện người dùng và kết hợp với module xử lý form (thường nằm ở máy server)



11.2 Tổng quát về lập trình Web chạy ở phía client

Các công nghệ lập trình Web chạy ở phía client khác nhau thường có các khuyết điểm sau :

- **không tin cậy** : phụ thuộc vào cấu hình máy client vốn rất đa dạng về công suất tính toán, tình trạng nhiễm virus...
- **không hiệu quả** : tốn thời gian download phần tử phần mềm từ server về client nhất là phần tử phần mềm có dung lượng lớn. Nếu dữ liệu xử lý nằm ở server thì phần mềm phải tốn nhiều thời gian truy xuất dữ liệu thông qua mạng.
- **khó bảo mật** : vì thành phần phần mềm phải được download về máy client nên client có thể tìm hiểu, nghiên cứu được thuật toán của phần mềm, nhất là nó được viết bằng ngôn ngữ script ở dạng rõ (mã nguồn).



11.2 Tổng quát về lập trình Web chạy ở phía client

- ❑ Với các yếu điểm của lập trình Web chạy ở phía client được miêu tả trong slide trước, người ta thường chỉ dùng nó để xử lý sơ bộ quá trình tương tác và nhập liệu của người dùng, còn để xử lý các vấn đề còn lại, người ta dùng các công nghệ lập trình Web ở Server.
- ❑ Trong phần còn lại của chương này, chúng ta sẽ trình bày 2 công nghệ lập trình Web để chạy ở máy Client dùng ngôn ngữ Java, đó là công nghệ Java Applet và công nghệ script.



11.3 Tổng quát về Java Applet

- ❑ Theo góc nhìn người dùng, Applet là 1 form giao diện với người dùng, ta có thể nhúng nó vào 1 container nào đó.
- ❑ Theo góc nhìn lập trình hướng đối tượng bằng Java, Applet được đặc tả bởi 1 class Java, class này phải là class con của class Applet. Bản thân class Applet lại là con của class Panel. Như vậy, nơi nào Panel tồn tại được thì nơi đó Applet cũng tồn tại được.
- ❑ Applet có thể nằm trong container nào đó (Panel, Frame,...). Tuy nhiên Applet được dùng chủ yếu trong các trang Web. Cả trang Web và Applet lúc đầu được xây dựng, kiểm thử và cài đặt ở máy Webserver, khi người dùng trên máy client yêu cầu, trang Web sẽ được download về máy client, chương trình xử lý trang Web ở máy client (thường là Web Browser) sẽ yêu cầu download applet và add applet vào trang Web container trước khi hiển thị cho người dùng xem và làm việc.



11.3 Tổng quát về Java Applet

Cấu trúc của 1 class Applet

```
public class MyApplet extends Applet
    implements KeyListener, MouseListener {
    //định nghĩa các thuộc tính cần dùng
    //override tác vụ đọc giá trị các tham số container truyền sang
    public String getParameter(String key, String def) {...}
    //override tác vụ khởi tạo các thuộc tính của Applet
    public void init() {...}
    //override tác vụ vẽ lại giao diện của Applet theo trạng thái hiện hành
    public void paint(Graphics g) {...}
    //hiện thực các hàm xử lý sự kiện về phím và về chuột theo yêu cầu
    //override các tác vụ khác & định nghĩa các hàm khác theo yêu cầu
}
```



11.3 Tổng quát về Java Applet

Viết và dịch Java Applet

- Thường mã nguồn của class Applet được đặt trên 1 file *.java. Nếu dùng cửa sổ hàng lệnh để dịch file *.java, ta dùng cú pháp sau :

javac MyApplet.java

- Sau khi dịch xong, máy sẽ tạo ra 1 file MyApplet.class, đôi khi còn có thêm file MyApplet\$1.class. Ta nên copy các file *.class miêu tả applet sang thư mục chứa trang Web sử dụng applet.
- Nếu dùng môi trường lập trình trực quan nào đó (JBuilder, NetBeans, Eclipse,...) thì ta tuân theo qui trình xây dựng Applet do môi trường đó qui định, nhưng nhìn chung gồm các bước : tạo Project quản lý Applet, tạo class Applet, hiện thực chi tiết Applet, dịch và kiểm thử Applet, copy các file *.class về thư mục chứa trang Web sử dụng.



11.3 Tổng quát về Java Applet

Đóng gói và sử dụng Applet

- Trong trường hợp Applet dùng nhiều file tài nguyên khác (class khác, file ảnh,...), ta nên tạo 1 file *.jar chứa applet và tất cả file tài nguyên mà nó sử dụng rồi copy file *.jar sang thư mục chứa trang Web sử dụng applet.

- Nhúng applet vào vị trí mong muốn trong trang Web dùng tag `<applet>` :

`<p>Dưới đây là MyApplet</p>`

`<applet code="MyApplet.class" width=138 height=190>`

`<param name="rong" value=138>`

`<param name="cao" value=190>`

`</applet>`

- Nếu applet nằm trong file *.jar, dùng thêm mệnh đề `archive="MyApplet.jar"` trong tag `<applet>`.



11.4 Qui trình điển hình để xây dựng và dùng Applet

- ❑ Để thấy rõ ràng và cụ thể việc xây dựng, kiểm thử và dùng Applet, ta hãy dùng NetBeans xây dựng một applet đơn giản có chức năng giải phương trình bậc 2. Cấu trúc của Applet cũng giống với form ứng dụng giải phương trình bậc 2 trong chương 6 : gồm 3 Label và 3 Textbox để nhập 3 tham số của phương trình, 1 Button để người dùng ra lệnh giải phương trình, 3 Label để hiển thị kết quả.
 1. Chạy NetBeans, nếu có Project cũ nào được mở tự động thì hãy đóng các Project này đi.
 2. Chọn menu File.New Project để máy hiển thị cửa sổ "New Project", chọn mục "Java" trong Listbox Categories, chọn mục "Java Applications" trong Listbox Projects rồi click button Next để hiển thị cửa sổ "New Java Application".



11.4 Qui trình điển hình để xây dựng và dùng Applet

3. Xác định thư mục chứa Project ở textbox "Project Location", nhập "GPTB2Applet" vào textbox "Project Name", click button Finish để máy tạo thực sự Project.
4. Máy đã tạo class miêu tả ứng dụng với tên mặc định là GPTB2Applet, mã nguồn của class này cũng đang được hiển thị, hãy viết lại code cho class này như sau để đặc tả Applet giải phương trình bậc 2 :

```
package gptb2applet;  
//import các package thư viện cần dùng  
import java.awt.*;  
import java.awt.event.*;  
import java.applet.Applet;
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

//đặc tả class applet

```
public class GPTB2Applet extends Applet {
```

//định nghĩa các thuộc tính của Applet

```
private Label labelA;
```

```
private TextField txtA;
```

```
private Label labelB;
```

```
private TextField txtB;
```

```
private Label labelC;
```

```
private TextField txtC;
```

```
private Button btnStart;
```

```
private Label lblKetqua;
```

```
private Label lblX1;
```

```
private Label lblX2;
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

//hàm giải phương trình bậc 2 theo 3 tham số nhập

```
void giaiPTB2() {
```

```
    //định nghĩa các biến cần dùng
```

```
    double a, b, c;
```

```
    double delta;
```

```
    double x1, x2;
```

```
    //mã hóa dữ liệu chuỗi thành giá trị số Double
```

```
    a = Double.parseDouble(txtA.getText());
```

```
    b = Double.parseDouble(txtB.getText());
```

```
    c = Double.parseDouble(txtC.getText());
```

```
    //tính biệt số delta
```

```
    delta = b * b - 4 * a * c;
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//kiểm tra biệt số delata và quyết định xử lý
if (delta >= 0) { //trường hợp có 2 nghiệm thực
    x1 = (-b + Math.sqrt(delta)) / 2 / a;
    x2 = (-b - Math.sqrt(delta)) / 2 / a;
    lblKetqua.setText("Phuong trinh co 2 nghiệm :");
    lblX1.setText("x1 = " + x1);
    lblX2.setText("x2 = " + x2);
} else { //trường hợp vô nghiệm
    lblKetqua.setText("Phuong trinh vo nghiệm !");
    lblX1.setText("");
    lblX2.setText("");
}
}
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//hàm khởi tạo trạng thái đầu cho applet
public void init() {
    //doc tham so moi truong va thiet lap kich thuoc applet
    //int r = Integer.parseInt(this.getParameter("rong"));
    //int c = Integer.parseInt(this.getParameter("cao"));
    //this.setSize(r, c);
    //thiết lập kích thước frame
    this.setSize(350,180);
    //thiết lập LayoutManager cho Frame
    this.setLayout(new GridLayout(7,1));
}
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Panel chứa label và textbox A
Panel mPanel = new Panel();
//khai báo layout cho Panel có 2 cột
mPanel.setLayout(new GridLayout(1, 2));
//tạo Label và add nó vào Panel
labelA = new Label("Nhập a : ");
mPanel.add(labelA);
//tạo Textbox và add nó vào Panel
txtA = new TextField("");
mPanel.add(txtA);
//add Panel vào Frame
add(mPanel);
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Panel chứa label và textbox B
mPanel = new Panel();
//khai báo layout cho Panel có 2 cột
mPanel.setLayout(new GridLayout(1, 2));
//tạo Label và add nó vào Panel
labelB = new Label("Nhập b : ");
mPanel.add(labelB);
//tạo Textbox và add nó vào Panel
txtB = new TextField("");
mPanel.add(txtB);
//add Panel vào Frame
add(mPanel);
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Panel chứa label và textbox C
mPanel = new Panel();
//khai báo layout cho Panel có 2 cột
mPanel.setLayout(new GridLayout(1, 2));
//tạo Label và add nó vào Panel
labelC = new Label("Nhập c : ");
mPanel.add(labelC);
//tạo Textbox và add nó vào Panel
txtC = new TextField("");
mPanel.add(txtC);
//add Panel vào Frame
add(mPanel);
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Panel có 3 cột để chứa Button ở giữa
mPanel = new Panel(); mPanel.setLayout(new GridLayout(1, 3));
mPanel.add(new Label(""));
//tạo Button và add nó vào Panel
btnStart = new Button("Bắt đầu giải");
//định nghĩa hàm xử lý click chuột trên Button
btnStart.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        giaiPTB2(); //gọi hàm giải ptb2
    }
});
mPanel.add(btnStart); //add Button vào Panel
mPanel.add(new Label(""));
add(mPanel); //add Panel vào Frame
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Panel chứa Label kết quả  
mPanel = new Panel();  
//khai báo layout cho Panel có 2 cột  
mPanel.setLayout(new GridLayout(1, 1));  
//tạo Label kết quả và add nó vào Panel  
lblKetqua = new Label("");  
mPanel.add(lblKetqua);  
add(mPanel); //add Panel vào Frame  
//tạo Panel chứa Label x1 =  
mPanel = new Panel();  
//khai báo layout cho Panel có 2 cột  
mPanel.setLayout(new GridLayout(1, 1));
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
//tạo Label x1= và add nó vào Panel
lblX1 = new Label("");
mPanel.add(lblX1);
add(mPanel); //add Panel vào Frame
//tạo Panel chứa Label x2 =
mPanel = new Panel();
mPanel.setLayout(new GridLayout(1, 1));
//tạo Label x2= và add nó vào Panel
lblX2 = new Label("");
mPanel.add(lblX2);
add(mPanel); //add Panel vào Frame
}
}
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

5. dờ chuột về mục gptb2applet trong cây Project, ấn kép chuột vào nó để hiển thị menu lệnh, chọn option New.Java class để hiển thị cửa sổ "New Java class", nhập tên "Program" vào textbox "Class name" rồi click button Finish để máy tạo thực sự class mới theo yêu cầu, class này miêu tả ứng dụng thử dùng Applet, nó tạo 1 frame, frame này chứa applet GPBT2Applet để ta có thể dịch và kiểm thử applet :

```
package gptb2applet;  
//import các package thư viện cần dùng  
import java.awt.*;  
import java.awt.event.*;
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
public class Program { //đặc tả class ứng dụng
    public static void main(String[] args) {
        //tạo frame giao diện
        Frame fr = new Frame();
        fr.setSize(350,180); fr.setLayout(new GridLayout(1,1));
        //tạo applet và add nó vào frame
        GPTB2Applet apl = new GPTB2Applet(); apl.init();
        fr.add (apl);
        //định nghĩa hàm xử lý sự kiện đóng Form
        fr.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) { System.exit(0); }
        });
        fr.setVisible(true);
    }
}
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

6. ấn phải chuột trên mục Program.java trong cây Project để hiển thị menu lệnh, chọn option “Run File” để dịch và chạy thử chương trình dùng Applet. Nếu có lỗi từ vựng và cú pháp thì sửa, nếu có lỗi run-time thì debug (thông qua các chức năng trong menu Debug) để xác định lỗi rồi sửa lỗi.
7. Sau khi kiểm thử và applet chạy tốt, hiển thị lại cửa sổ mã nguồn của applet, và hiệu chỉnh các lệnh ở đầu hàm init như sau :

```
public void init() {  
    //đọc tham số môi trường & thiết lập kích thước applet  
    int r = Integer.parseInt(this.getParameter("rong"));  
    int c = Integer.parseInt(this.getParameter("cao"));  
    this.setSize(r, c);  
    //this.setSize(230,180);  
}
```



11.4 Qui trình điển hình để xây dựng và dùng Applet

8. Chọn menu Run.Clean and build Main Project để dịch lại và tạo lại file *.jar chứa kết quả.
9. copy file dist\GTPB2Applet.jar do chương trình dịch tạo ra về thư mục chứa trang Web dùng applet.
10. Chạy chương trình soạn văn bản ưa thích (NotePad, WordPad, Word,...) rồi soạn nội dung 1 trang Web đơn giản như sau để thử kiểm tra Applet :



11.4 Qui trình điển hình để xây dựng và dùng Applet

```
<html>
<head>Trang Web dùng Applet</head>
<body>
<p>Đây là Applet giải phương trình bậc 2, bạn hãy thử dùng nó
:</p>
<applet code="gptb2applet.GPTB2Applet"
        archive="GPTB2Applet.jar" width=250 height=180>
<param name="rong" value=250>
<param name="cao" value=180>
</applet>
</body>
</html>
```

11. Lưu nội dung trang Web lên file UseGPTB2.htm, file này nằm ở thư mục chứa file *.jar miêu tả applet. Thử dùng IE truy xuất trang Web và dùng thử Applet.



11.5 Viết và dùng java script

Java script là ngôn ngữ kịch bản, nó là tập con của Java. Javascript được dùng trong các trang Web để tăng cường chức năng của trang Web. Một số điểm khác biệt chính yếu giữa Javascript và Java là :

- JavaScript được thông dịch bởi Web browser hay Web Server, còn Java được dịch trước khi chạy (ở client hay server).
- Javascript chỉ dùng các đối tượng có sẵn, còn Java cho phép định nghĩa mới các class dựa trên cơ chế thừa kế.
- Javascript được viết trong trang web, còn Java được viết trên các thành phần độc lập như applet, servlet, javabeen...
- Javascript không kiểm tra kiểu, còn Java kiểm tra kiểu chặt chẽ.
- Javascript kiểm tra động tham khảo đến đối tượng được dùng, Java kiểm tra tham khảo tại thời điểm dịch...



11.5 Viết và dùng java script

- ❑ Ta nhúng đoạn code javascript vào trang web bằng tag lệnh :
 <script LANGUAGE=javascript>
 //đoạn code javascript
 </script>
- ❑ Mỗi trang web có thể chứa nhiều tag <script>, mỗi tag script có thể chứa các lệnh javascript để định nghĩa :
 - các biến toàn cục được dùng cho toàn trang web.
 - các hàm chức năng được dùng trong trang web.
- ❑ Thường các hàm javascript có thể là :
 - hàm dịch vụ được gọi từ các hàm khác.
 - Hàm xử lý sự kiện được kích hoạt từ người dùng.



11.5 Viết và dùng java script

- ❑ Trang Web chứa nhiều tag lệnh, mỗi tag lệnh thường miêu tả 1 phần tử được hiển thị nào đó, thí dụ như đoạn văn, ảnh bitmap, button, textbox, listbox,...
- ❑ Mỗi phần tử được hiển thị được gọi là phần tử giao diện, nó đáp ứng được nhiều sự kiện mà người dùng tạo ra trên nó như ấn/thả phím, ấn/thả nút chuột :

onafterupdate :	xảy ra sau khi phần tử đã được cập nhật (database)
onbeforeupdate :	trước khi phần tử đã được cập nhật (database)
onclick :	sau khi ấn chuột trên phần tử
ondblclick :	sau khi ấn kép chuột trên phần tử
ondragstart :	khi bắt đầu drag phần tử
onerrorupdate :	khi có sai trong việc cập nhật phần tử
onhelp :	khi có yêu cầu trợ giúp



11.5 Viết và dùng java script

onkeydown :	khi ấn phím
onkeypress :	khi ấn phím
onkeyup :	khi nhả phím
onmousedown :	khi ấn mouse
onmousemove :	khi dời mouse
onmouseout :	khi dời mouse ra khỏi phần tử
onmouseover :	khi dời mouse tới phần tử
onmouseup :	khi nhả mouse
onreadystatechange :	khi đối tượng thay đổi trạng thái
onrowenter :	khi record hiện hành đã thay đổi nội dung
onrowexit :	khi điều khiển datasource thay đổi nội dung record.
onselectstart :	khi đối tượng được chọn.



11.5 Viết và dùng java script

- ❑ Người viết trang web có quyền khai báo hàm xử lý sự kiện cho sự kiện xảy ra trên phần tử giao diện mong muốn theo cú pháp :

`<p onclick=func1 ()>Thử click chuột trên chuỗi này </p>`



11.6 Thí dụ viết trang Web có dùng java script

- ❑ Chúng ta hãy xây dựng 1 Website phục vụ trò chơi dò mìn đơn giản có trang homepage như sau (trạng thái đang chơi) :

Trang web demo tro chơi do min

Ngươi viet : Nguyen Van Hiep

1	1	1					
1		1					
1	2	3	3	3	2	1	
	2					1	
	2				3	1	
1	2	2	4		2		
	1		1		1		

Hay do tung o min



11.6 Thí dụ viết trang Web có dùng java script

- Trang homepage ở trạng thái thua như sau :

Trang web demo tro chơi do min

Người viết : Nguyen Van Hiep

	1	2	2	1	1	★	
	1	★	★	2	2		
	1	2	3	★	2		
			1	1	2	★	
		1	2	2	2		
1	1	1	★	★	2		★
★	1	1	2	3	★		
1	1			1			

Chơi lại ván khác

Ba da chet



11.6 Thí dụ viết trang Web có dùng java script

- ❑ Phần tử chính của trang web dò mìn là ma trận mìn, thí dụ ta chọn kích thước cố định là 8×8 . Để thể hiện ma trận mìn, ta dùng 64 tag ``, chia làm 8 nhóm, mỗi nhóm 8 tag `` miêu tả 1 hàng mìn được kết thúc bởi tag `
` để xuống hàng. Mỗi tag `` thể hiện 1 ô mìn, mỗi ô mìn có các trạng thái sau :
 - chưa đập (thể hiện bằng ảnh `wm_nul.gif`)
 - đã đập (thể hiện bằng 1 trong 11 ảnh bitmap : 0-8 mìn xung quanh nó, mìn nổ, mìn chưa kịp nổ).
- ❑ Dữ liệu chính của trò chơi gồm 2 biến array :
 - `status[8][8]`, phần tử `status[i][j]` chứa giá trị luận lý miêu tả cell tương ứng đã bị đập chưa.
 - `min[8][8]`, phần tử `min[i][j]` chứa giá trị luận lý miêu tả cell tương ứng có mìn hay không.



11.6 Thí dụ viết trang Web có dùng java script

- ❑ Về việc xử lý sự kiện, ta có 2 phương án khác nhau :
 - kết hợp toàn trang web với hàm xử lý sự kiện onclick, hàm này phải kiểm tra vị trí ấn chuột có nằm trong bàn mìn không, nếu có thì xác định vị trí i,j rồi kiểm tra có mìn không để đáp ứng phù hợp.
 - kết hợp với từng cell mìn 1 hàm xử lý sự kiện, hàm này xử lý ngay vị trí i,j để đáp ứng phù hợp theo trạng thái của vị trí đó.
- ❑ Các hàm chức năng :
 - `domin(i,j)` kiểm tra việc đạp vào vị trí i,j .
 - `doquanh(i,j)` đi dùm người dùng các cell quanh cell (i,j) vì biết chắc không có mìn.
 - `count(i,j)` đếm số mìn xung quanh vị trí (i,j) .



11.6 Thí dụ viết trang Web có dùng java script

- ❑ Ta có thể dùng trình soạn thảo bitmap như Paint để soạn thảo từng hình theo yêu cầu. Lưu ý tất cả hình bitmap đều có cùng kích thước (thí dụ 17*17).
- ❑ Tạo thư mục miêu tả webiste, copy tất cả các file hình vừa soạn vào thư mục này, rồi dùng trình soạn thảo văn bản bất kỳ để soạn nội dung trang web trò chơi như các slide kế.
- ❑ Lưu ý nội dung chính của trang Web gồm 2 phần : phần HTML miêu tả ma trận mìn dùng 64 tag miêu tả ma trận 8*8 ô, phần code java script miêu tả các hàm xử lý ấn chuột và các hàm dịch vụ kèm theo



11.6 Thí dụ viết trang Web có dùng java script

```
<html><head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<title></title>
<script LANGUAGE=javascript>
//Định nghĩa các biến toàn cục
var odado=0;
var dangchoi=0;
var data_tbl;
var stat_tbl;
//hàm khởi tạo mảng động
function MakeArray(size) {
    this.length = size;
    return this;
}
```



11.6 Thí dụ viết trang Web có dùng java script

//hàm khởi tạo trò chơi dò mìn

```
function wm_init() {
```

```
var i,j;
```

```
    odado = 0; dangchoi = 1;
```

```
    //hiển thị bàn mìn trống
```

```
    for (i =0; i<64; i++)
```

```
        document.images[i].src = "wm_nul.gif";
```

```
    //hiển thị chuỗi cảnh báo và ẩn button “Chơi tiếp”
```

```
    document.all.Mesg.innerText = "Hãy dò từng ô mìn";
```

```
    document.all.IDSTART.style.display = "none";
```

```
    //tạo các bảng dữ liệu và thiết lập giá trị đầu.
```

```
    data_tbl= new MakeArray(64);
```

```
    stat_tbl= new MakeArray(64);
```

```
    for (i=0; i<64; i++) data_tbl[i] = stat_tbl[i] = 0;
```



11.6 Thí dụ viết trang Web có dùng java script

```
//xếp ngẫu nhiên 10 trái mìn  
i = 0;  
while (i <10) {  
    j = Math.round(Math.random()*62);  
    if (data_tbl[j]==0) { data_tbl[j]=1; i++; }  
}  
}
```



11.6 Thí dụ viết trang Web có dùng java script

//hàm tính số mìn xung quanh vị trí (h,c)

```
function winecnt (h, c) {  
    var cnt =0;  
    if (h-1>=0 && c-1>=0 && data_tbl[(h-1)*8+(c-1)]) cnt++;  
    if (h-1>=0 && data_tbl[(h-1)*8+c]) cnt++;  
    if (h-1>=0 && c+1<8 && data_tbl[(h-1)*8+(c+1)]) cnt++;  
    if (c-1>=0 && data_tbl[h*8+(c-1)]) cnt++;  
    if (c+1<8 && data_tbl[h*8+(c+1)]) cnt++;  
    if (h+1<8 && c-1>=0 && data_tbl[(h+1)*8+(c-1)]) cnt++;  
    if (h+1<8 && data_tbl[(h+1)*8+c]) cnt++;  
    if (h+1<8 && c+1<8 && data_tbl[(h+1)*8+(c+1)]) cnt++;  
    return cnt;  
}
```



11.6 Thí dụ viết trang Web có dùng java script

//hàm click dùm tự động các ô bao quanh vị trí (h,c)

```
function doquanh(h,c) {  
    if (h-1>=0 && c-1>=0 && stat_tbl[(h-1)*8+(c-1)]==0) domin(h-1,c-1);  
    if (h-1>=0 && stat_tbl[(h-1)*8+c]==0) domin(h-1,c);  
    if (h-1>=0 && c+1<8 && stat_tbl[(h-1)*8+(c+1)]==0) domin(h-1,c+1);  
    if (c-1>=0 && stat_tbl[h*8+(c-1)]==0) domin(h,c-1);  
    if (c+1<8 && stat_tbl[h*8+(c+1)]==0) domin(h,c+1);  
    if (h+1<8 && c-1>=0 && stat_tbl[(h+1)*8+(c-1)]==0) domin(h+1,c-1);  
    if (h+1<8 && stat_tbl[(h+1)*8+c]==0) domin(h+1,c);  
    if (h+1<8 && c+1<8 && stat_tbl[(h+1)*8+(c+1)]==0) domin(h+1,c+1);  
}
```



11.6 Thí dụ viết trang Web có dùng java script

//hàm xử lý việc click vào ô (h,c)

```
function domin(h,c){
```

```
var i,cnt;
```

```
    i = h*8+c;
```

```
    if (stat_tbl[i]) return 0; //đã đập rồi ô này
```

```
    if (data_tbl[i] ==1) { //đập trúng mìn
```

```
        for (h=0;h<64;h++) {
```

```
            if (data_tbl[h]==1) document.images[h].src = "wm_mini.gif";
```

```
        }
```

```
        document.images[i].src = "wm_mina.gif";
```

```
        document.all.Mesg.innerText = "Bạn đã chết";
```

```
        document.all.IDSTART.style.display = "";
```

```
        dangchoi = 0;
```

```
    } else { //dò ô không có mìn
```



11.6 Thí dụ viết trang Web có dùng java script

```
cnt = winecnt(h,c); stat_tbl[i] = 1;
if (cnt ==0) {
    document.images[i].src = "wm_nul1.gif"; doquanh(h,c);
}
if (cnt ==1) document.images[i].src = "wm_1.gif";
if (cnt ==2) document.images[i].src = "wm_2.gif";
if (cnt ==3) document.images[i].src = "wm_3.gif";
if (cnt ==4) document.images[i].src = "wm_4.gif";
if (cnt ==5) document.images[i].src = "wm_5.gif";
if (cnt ==6) document.images[i].src = "wm_6.gif";
if (cnt ==7) document.images[i].src = "wm_7.gif";
if (cnt ==8) document.images[i].src = "wm_8.gif";
odado++;
}
```



11.6 Thí dụ viết trang Web có dùng java script

```
if (odado ==54) {  
    document.all.Mesg.innerText = "Bạn đã thắng";  
    document.all.IDSTART.style.display = "";  
    dangchoi = 0;  
}  
}  
//hàm xử lý ấn chuột trên trang Web  
function document_onmousedown() {  
var i,h;  
    if (dangchoi == 0) return 0;  
    //xác định vị trí luận lý của phần tử được user ấn chuột  
    i = window.event.srcElement.sourceIndex-12;  
    if (i<0) return 0;
```



11.6 Thí dụ viết trang Web có dùng java script

//xác định tọa độ h,c của cell mình được user ấn chuột

c = i%9; h = (i-c) /9;

if (h >=8 || c == 8) return 0;

domin(h,c); //xử lý việc ấn cell h,c

}

//hàm xử lý click chuột trên button Start

function btnStart_onclick() { wm_init(); }

</script>

</head>

<body onload="wm_init()" onclick="document_onmousedown()">

<center>

<H1 style="color:red">Trang web demo trò chơi dò mìn</H1>

<h2 style="color:blue">Người viết : Nguyễn Văn Hiệp</h2>

<p></p>



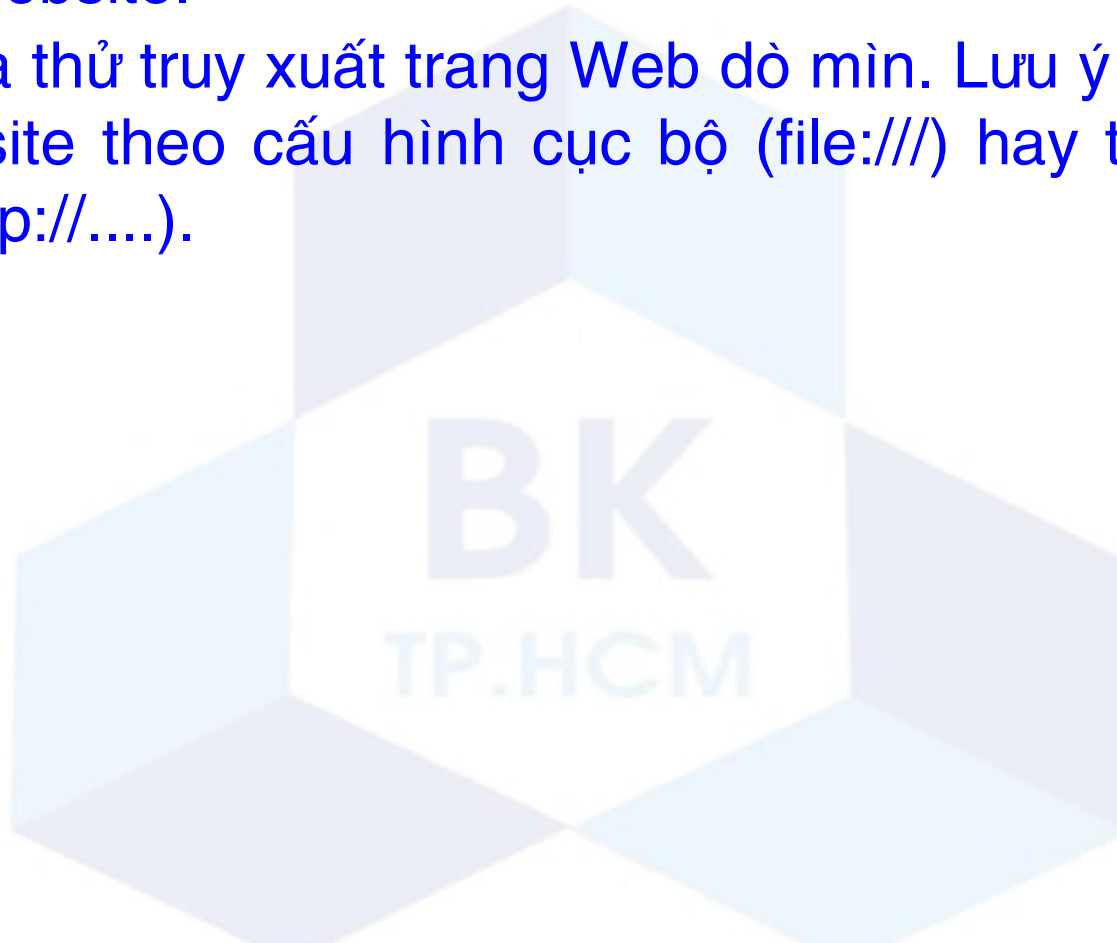
11.6 Thí dụ viết trang Web có dùng java script

```
 ... <br>
 ... <br>
 ... <br>
 ... <br>
 ... <br>
 ... <br>
 ... <br>
 ... <br>
<p><input id=IDSTART style="display:none" type="button"
    value="Chơi lại ván khác" language=javascript
    onclick="btnStart_onclick()"></p>
<h2 ID="Mesg" STYLE="color:magenta">Hãy dò từng ô mìn</h2>
</center>
</body></html>
```



11.6 Thí dụ viết trang Web có dùng java script

- ❑ Sau khi soạn xong trang Web, cất nó lên file WinMine.html trong thư mục Website.
- ❑ Chạy IE và thử truy xuất trang Web dò mìn. Lưu ý bạn có thể truy xuất Website theo cấu hình cục bộ (file:///) hay thông qua Web Server (http://.....).



11.7 Kết chương

- ❑ Chương này đã trình bày kiến thức về lập trình Web chạy ở phía client dùng ngôn ngữ Java và Javascript.
- ❑ Chương này đã giới thiệu công dụng của Applet, cấu trúc tổ chức của Applet, cách viết, kiểm thử 1 applet trước khi đóng gói nó vào file *.jar để phân phối cho website sử dụng nó.
- ❑ Chương này cũng đã giới thiệu cách viết trang Web có sử dụng Applet, cách viết tag <applet> để đặc tả thông tin về applet sử dụng.
- ❑ Chương này cũng đã trình bày các tính chất của Javascript, công dụng của nó và giới thiệu 1 thí dụ về cách viết trang web có dùng các đoạn lệnh Javascript.

