



Chapter 9

Module7



MCQ Test



- ✓ Closed book
- ✓ Closed computer
- ✓ 20 questions
- ✓ Leave if finished, but come back at 1:00 for lecture



MCQ Test



✗ Any questions?



Java



- ✓ No more lecture material, but
- ✓ I will respond to questions with material as needed
- ✓ Following few weeks will have Java/Swing centered questions, and
- ✓ Assignment 3 (announced next week) will require Java/Swing, so get some practice in...



Common Gateway Interface



- ✓ CGI is a standard for helping web servers run *external* programs,
- ✓ and return *dynamic* web pages.

For example, a simple dynamic web page might return the current date and time, calculated by running the 'date' program, and formatting the results as a web page.



CGI script



```

CODE LISTING          mydate.cgi
-----
#!/bin/sh
cat <<BOM1
Content-type: text/html

<HTML><HEAD>
<TITLE>Output of data in HTML from CGI script</TITLE>
</HEAD><BODY>
<H1>Date:</H1>
BOM1
date
cat <<BOM2
</BODY></HTML>
BOM2

```



CGI script

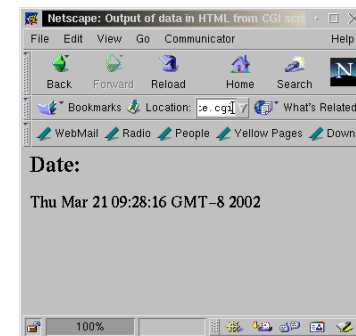


When this script is placed in the directory `public_cgi` in your home directory on one of the UNIX systems, then you may refer to

<http://www-cgi.comp.nus.edu.sg:8000/~yourid/mydate.cgi>



CGI script





CGI script



```

NetScape: Source of: http://www.cgi.comp.nus.edu.sg/3000/~hugh/mydate.cgi
<HTML><HEAD>
<TITLE>Output of data in HTML from CGI script</TITLE>
</HEAD><BODY>
<H1>Date:</H1>
Thu Mar 21 09:28:16 GMT-8 2002
</BODY></HTML>

```



CGI scripts



- ✓ No requirement for CGI program to be a shell script.
- ✓ **Perl** is very commonly used in this role.
- ✓ It should not take too long to process.



Environment variables



```

CODE LISTING          env.cgi
#!/usr/local/bin/perl
print "Content-type: text/html\n\n";
print <<EndOfHTML;
<html><head><title>Print Environment</title></head>
<body>
EndOfHTML

foreach $key (sort(keys %ENV)) {
    print "$key = $ENV[$key]<br>\n";
}

print "</body></html>";

```



Environment variables



```

DOCUMENT_ROOT = /usr/local/apache/htdocs
GATEWAY_INTERFACE = CGI/1.1
HTTP_USER_AGENT = Mozilla/4.79 [en] (X11; U; Linux 2.2.16 i686)
...
QUERY_STRING =
...
TZ = Singapore

```



CGI forms - GET



The form contents are found inside an environment variable called `QUERY_STRING`, as a series of `name/value` pairs.

This mechanism is known as the `GET` mechanism, and a typical URL would look like this:

```
.../myform.cgi?name1=value1&name2=value2
```



POST



An alternative mechanism is the `POST` mechanism, in which the `STDIN` of the CGI program is used to process the form data.



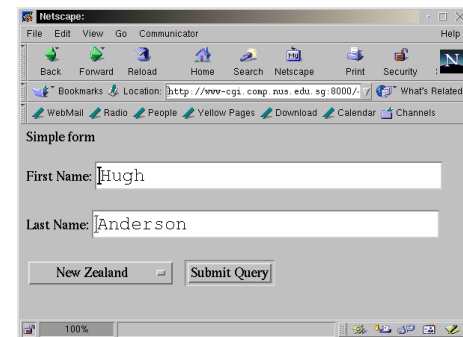
CGI form



CODE LISTING	form.html
<pre> <html><head>Simple form</head> <body> <form action="/env.cgi" method="GET"> First Name: <input type="text" name="First" size=30><p> Last Name: <input type="text" name="Last" size=30><p> <select name="Home"> <option>Singapore <option>Malaysia <option>Indonesia <option>New Zealand <option>The rest of the world; </select> <input type="submit"> </form> </body></html> </pre>	



CGI form





CGI form



When the form is submitted, the `QUERY_STRING` looks like this:

```
QUERY_STRING = First=Hugh&Last=Anderson&Home=New+Zealand
```

Within a CGI program, this series of name-value pairs may be used to return a dynamic web page based on this form data.

`Perl` is a particularly useful language to use in this context - the `QUERY_STRING` can be `split` quickly into its component parts.



Security



- ✓ There are security issues with unrestricted CGI programs - since they run powerful programs (like perl and csh) with arbitrary parameters, they may be a source of (hacker) intrusion.
- ✓ It is for this reason that CGI usage is restricted here at NUS.



PHP



- ✓ PHP is a server-side scripting language.
- ✓ It looks very like standard HTML scripts, but the server automatically interprets the PHP.
- ✓ There are no enhancements needed for browsers.



PHP



- ✓ The two tags `<?php` and `?>` start and end a PHP script, and identify a PHP code segment.
- ✓ The PHP code itself is a reasonably powerful programming language similar to Java, C and Perl, with functions, variables and so on.

PHP stands for PHP - Hypertext Preprocessor, a recursive acronym.



PHP



- ✓ Particularly useful to access databases.
- ✓ It is common to pair up PHP with MySQL, but PHP is not limited to one database type.
- ✓ For example if you wish to use PHP to access a Microsoft SQL server, you can install the ODBC support in the server machine, and access the server directly.



PHP code



```
<?php
...
mysql_pconnect("host", "user", "password")
  or die("Unable to connect to SQL server");
mysql_select_db("dbname")
  or die("Unable to select database");
$numguests = mysql_query("SELECT COUNT(*) FROM guests")
  or die("Select Failed!");
...
?>
```



PHP security



- ✓ PHP suffers less from the security issue than perl or csh CGI scripts do.



Java



```
CODE LISTING                               Lissajous1.java
/*
 * Lissajous1.java
 * Copyright 1999 was written in 0.4.86/04/09
 * by Hugh Anderson for NetJava browser.
 * Updated by L. Gladney to Java 1.0 JDK on 4/13/97.
 * Patrick Chan (chan@cs.cmu.edu) has suggested that it
 * could be nice if every point had a different display, so mouse
 * controls the contrast, the ratio of frequencies, and mouse Y motion
 * controls the amplitude.
 */
import java.awt.*;
import java.applet.*;

public class Lissajous1 extends Applet implements Runnable {
    double x,y,z;
    int   E=100;
    int   dir=0;
    int   amp=50;
    int   phase=0;
    int   rev=0;
    int   x0=0, x1=360;
    int   y0=0, y1=360;
    int   z0=0, z1=100;

    public void paint(Graphics g) {
        int x,y,z;
        int dx=10, dy=10, dz=10;
        int x0=x1-dx, x1=x1+dx;
        int y0=y1-dy, y1=y1+dy;
        int z0=z1-dz, z1=z1+dz;
        for (x=x0; x<=x1; x+=dx) {
            for (y=y0; y<=y1; y+=dy) {
                for (z=z0; z<=z1; z+=dz) {
                    g.drawString(""+z, x+y*100, z);
                }
            }
        }
        phase = 0;
        if (phase >= 360) phase = 0;
        if (phase <= 0) phase = 360;
        dir = dir + 1;
    }
}
```



Java



```

CODE LISTING                               Lissajous2.java
public void run() {
    while (true) {
        animate();
        try { Thread.currentThread().sleep(delay); // delay
        } catch (Exception e) { }
    }
}

public void start() {
    if (animate == null) {
        animate = new Thread(this);
        animate.start();
    }
}

public void stop() {
    if (animate != null) {
        animate.stop();
        animate = null;
    }
}

public boolean mouseDown(Event e, int x, int y) {
    Graphics g;
    g = getGraphics();
    dXEN = e.x;
    System.out.println("Got a mouse event at " + x + ", " + y);
    return true;
}

public boolean mouseDrag(Event e, int x, int y) {
    if (e.getStateMask() & MouseEvent.BUTTON1_MASK) {
        amp = 0; ( Ek = 1 );
    }
    return true;
}

public String getRepeatInfo() {
    return "Lissajous by Hugh Anderson/Lary Gladney ";
}

public String[] getParameterInfo() {
    String [] info = {
        {"delay", "-int", "-delay, default=50"}
    };
    return info;
}

```



Java



This code may be found at

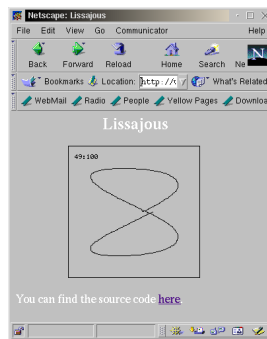
<http://oldept.physics.upenn.edu/courses/gladney/minicourse/lectures/lecture2.html>

or locally at

<http://www.comp.nus.edu.sg/~hugh/Lissajous/Lissajous.html>



Java



Summary of topics



In this module, we introduced the following topics:

- Web-based application architectures
- CGI, PHP and Java