

Web documents types

Three basic types of web documents

Static.

A static web document resides in a file that it is associated with a web server. The author of a static document determines the contents at the time the document is written. Because the contents do not change, *each request for a static document results in exactly the same response.*

Dynamic.

A dynamic web document does not exist in a predefined form. When a request arrives the web server **runs an application program** that creates the document. The server returns the output of the program as a response to the browser that requested the document. Because a fresh document is created for each request, *the contents of a dynamic document can vary from one request to another.*

Active

An active web document consists of a **computer program** that the server sends to the browser and that the browser must run locally. When it runs, the active document program can interact with the user and change the display continuously.

Advantages and disadvantages of each document type

Static

Advantages: simplicity, reliability and performance. The browser can place a copy in a cache on a local disk.

Disadvantages : inflexibility, changes are time consuming because they require a human to edit the file.

Dynamic

Advantages : ability to report *current information* (current stocks prices, current weather conditions, current availability of tickets for a concert).

Because both static and dynamic documents use HTML, a browser does not know whether the server extracted the page from a disk file or obtained the page dynamically from a computer program.

Disadvantages : increased cost and , like a static document, a dynamic document does not change after a browser retrieves a copy. *Thus , information in a dynamic document begins to age as soon as it as been sent to the browser(stock prices).*

Server push. The server runs the programs periodically and sends the new document to the browser

Active

Advantages : ability to update information continuously. For example, only an active document can change the display quickly *enough to show an animated image*. More important, an active document can access sources of information directly and update the display continuously. For example, an active document that displays stock prices can continue to retrieve stock information and change the display without requiring any action from the user.

Disadvantages :because an active document can run on an arbitrary computer instead on a server, the program must be written to avoid depending by particular features of a computer.

An active document is a potential risk because the document can export or import information.

Active documents require more sophisticated browser software (interpreters, virtual machines,..) and a powerful computer system to run the browser.

The active documents are normally written in source code. The compiler produces an executable form that is sent to the browser.

In the case of Java programs the source code is translated in the bytecode format, sent to the browser and locally executed by the java interpreter (JVM).

Implementation of dynamic documents

- 1) The server program must be extended so it is capable of executing a **separated application program** that creates a document each time a request arrives.
- 2) A separated application program must be written **for each dynamic document**.
- 3) The server must be configured so it knows which URLs correspond to dynamic documents and which correspond to static documents. For each dynamic document, the configuration must specify the application program that generates the document.

CGI (Common Gateway Interface)

- A widely used technology for **building dynamic Web documents** is known as the *Common Gateway Interface(CGI)*, originally developed by *the National Center for Supercomputer Applications (NCSA)*.
- The CGI standard specifies *how a server interacts* with an application program that implements a dynamic document. The application is called a **CGI program**.
- CGI does not specify a particular programming language. A programmer can choose an appropriate language for each document(C,C++, Perl, shell di Unix..).
- The CGI programs are placed in a directory called *bin*.
- The output of a CGI program may be a HTML program, but the standard permits CGI applications to generate arbitrary document types (plain text or a digital image).
The standard allows to place a header that describes the document type.

- After it runs a CGI program, a server examines the header before returning the document to the browser that issued the request.

For example, a header that consists of the line:

```
content- type: text/html
```

Followed by a blank line specifies that the output is a HTML document

The server sends the HTML documents to the browser by using HTTP protocol.

Parameters

A CGI program can be parameterized. A server can pass arguments to the program whenever the program is invoked.

A single program can settle a set of dynamic documents that differs only in minor details.

Values for the parameters can be supplied by the browser by inserting a suffix in the URL sent to the server

When a request arrives, the server divides the URL in the request into two parts: a prefix that specifies a particular document and a suffix that contains additional information.

If the prefix of the URL corresponds to a CGI program, the server invokes the program and passes the suffix of the URL as an argument.

Syntactically, a question mark (?) separates the prefix from the suffix

<http://www.unibo.it/dida/cgi-bin/orario/?giorno=010901&corso=C5>

Server-side scripting technologies

- A CGI program must generate an entire page, even if a few lines of HTML differ for each generation. In many instances, the bulk of a dynamic page remains the same for each occurrence. (stock quote, only the company name and current stock price need to be inserted dynamically; the heading and format information always remain the same).
- The server has a built-in interpreter that can make small modifications to a page as needed.
- The stored form of the page, which is known as a template or skeleton, contains a mixed of conventional HTML and scripting information.
- The interpreter allows conventional HTML to pass through unchanged, and replaces the scripting information with the results of interpreting the script

Server side scripting technologies

ASP (Active Server Pages). Dynamic page technology from Microsoft . The scripting information is written in the *Visual Basic* and the interpreter is closely integrated with Microsoft web server, *Internet Information Server* (IIS)

```
<html><body>  
<script language="vbscript" runat="server">  
For i=1 To 10  
Response.Write i & " "  
Next  
</script>  
</body></html>
```

It is called by the client through the URL:

<http://www.unserver.it/unadir/elenco.asp>

The result , server side, is:

```
html<body>  
1 2 3 4 5 6 7 8 9 10  
</body></html>
```

The HTML page is sent to the browser.

The code could be executed client- side if a visual basic interpreter is present on the browser.

JSP (Java Server Pages)

Is a dynamic page technology that it intended to be platform-independent Pages contain embedded scripting cod written in the Java programming language.

PHP (Perl Helper Pages)

Perl programming language.

Coldfusion

It is a dynamic page thecnology used to embed SQL database queries in pages. When a server handles such a page, the interpreter sends each SQL query to a database system, converts the result to HTML and replaces the query

```
<html><body>  
<h1> a dynamic document generated in JavaScript</h1>  
<script language="javascript">  
  for( i=1 ; i<=10; i++1)  
    document.write (i+ " ");  
</script>  
</body></html>
```

Applet

Java uses the term **applet** to describe active document program and to distinguish active documents from conventional computer programs.

Ex:

```
import java.applet,*;  
import java.awt,*;
```

```
public class contaclick extends Applet {  
    int counter ;  
    Textfield;  
    public void init( ) {  
        counter=0;        ;  
        add (new Button (“clicca qui”));  
        text= new TextField (*il pulsante non è stato ancora premuto”);  
        text.setEditable(false);  
        add (testo);  
    }  
}
```

```
public boolean action (Event e, Object arg) {  
    if ((Button e.target).getLabel() == "Clicca qui") {  
        counter+=1  
        text.setText(*il pulsante è stato premuto"+ contatore+"volte.");  
    }  
    return true;  
}  
}
```


Applet execution

<http://www.inesistente.edu/esempio/bbb.class>

bbb.class represents a file that must be acquired from the server and executed on the browser.

An applet tag is inserted on a HTML document. The tag is constituted of two parts, respectively called **codebase** and **code** .

```
<applet codebase= www.inexistent.edu/example code= "bbb.class">.
```

When the browser reads the applet tag, calls the server and a copy of the file *bbb.class* is moved to it. Then an object of the bbb class is created and its init method is called.