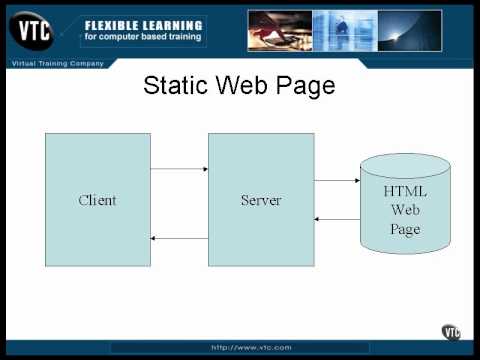
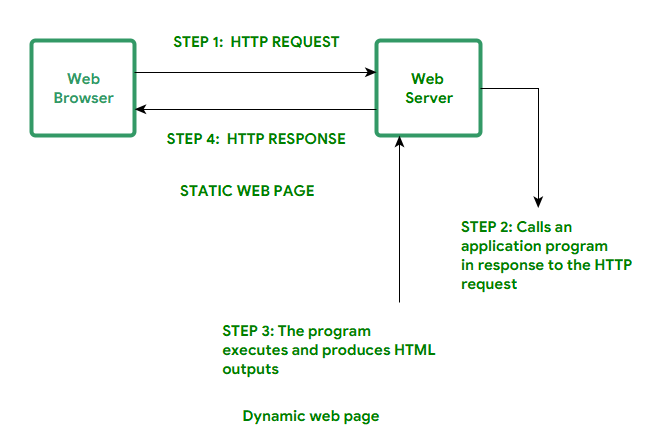
**DIFFERENCE BETWEEN STATIC AND DYNAMIC WEB PAGES**

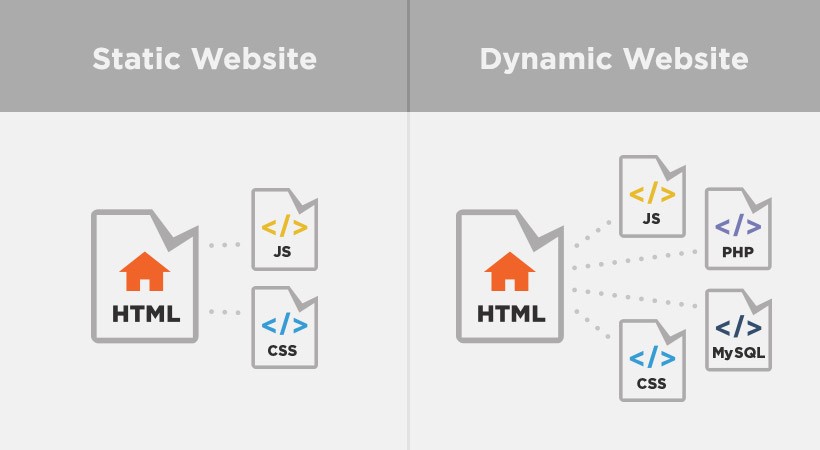
**STATIC WEB PAGES:** Static web pages are the web pages that are developed by using the languages like HTML, CSS, JavaScript etc. When an http request came to the server for a web page then the server immediately approves and gives the requested resources or webpages to the client without doing any additional processes. These web pages are seen through our web browsers like chrome and Firefox etc. These are web pages will remain constant or same until it will be changed manually.

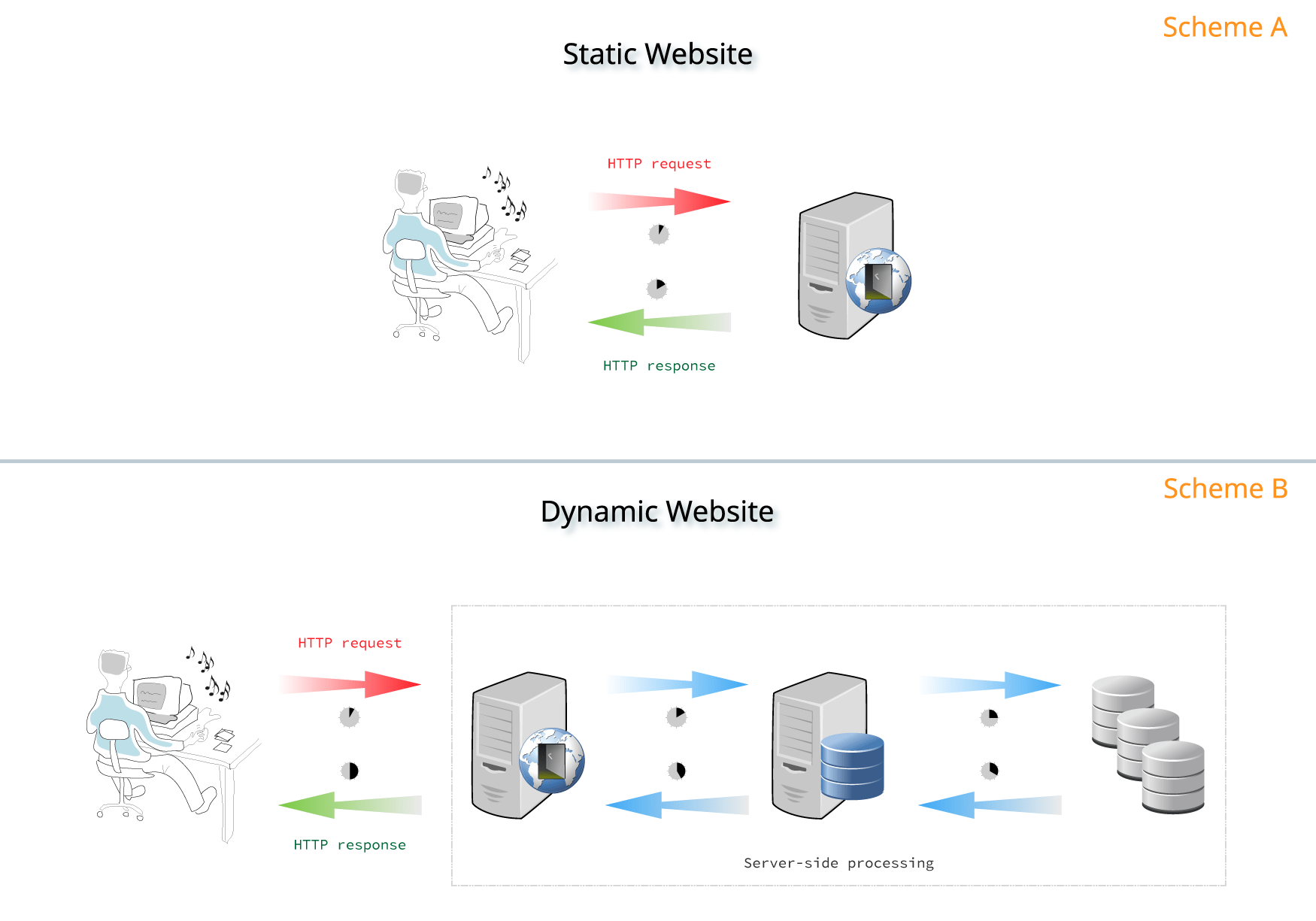


**DYNAMIC WEB PAGES:** Dynamic web pages are the web pages that are developed by using the languages like AJAX, CGI, ASP, ASP.NET etc. Dynamic web pages will not open same to all the users. The content in those web pages changes frequently. They are developed by using Machine Learning and Artificial Intelligence. So, the web pages will be responded differently to the different users. Example of Dynamic web pages are Weather detector, stock prices, Prediction apps etc.



**Pictorial difference between the static and dynamic web pages**





**SERVLETS:** Servlets are nothing but a simple java programs.

**WORKING OF SERVLETS:**

**Diagram

Description automatically generated**

As the web server cannot handle or develop the dynamic web pages, they use servlets to develop the dynamic web pages. And the web server cannot directly access the servlets. The web server access the servlets with the help of web container. When client request for any dynamic web page then the web server accepts the http request from client, and it will request the web container. This web container takes the http request from the web server and converts into a valid request object and sends to the servlet. Here the web container maintains a **web.xml** file to know what type of request is requested by the client and what type of servlet must be invoked according to the client’s request. According to the web.xml file the http request from the web server will converts into a valid request object and sends to the servlet.

As we know servlets are the simple java programs. So, the java programs are written in pure object-oriented programming way, and they are accessed only by using objects only. So, web container converts the request into object. Here the web container also creates a thread for each client request so that the multithreading concept can be implemented, and it will be faster way to give response to the multiple clients who are accessing the web page.

The servlet checks the client request, and it will make an appropriate response to that request by using call back methods. The web container responds thread for each request. It calls some call back methods like service() and doGet(). The doGet() contains valid business logic. The obtained servlet gathers the relevant information in order to satisfy the clients request and builds a response object and then sends it to the web container. The web container converts the given response object into a html or relevant response, because the web server cannot access the object and it cannot display them to the client. And it gives the http response to the web server and the web server gives the http response to the client. This is how the working of servlet will go through.

**LIFE CYCLE OF A SERVLET:**

Let us know the life cycle of a servlet. There are 3 methods in servlets. They are:

1. init()
2. service()
3. destroy()

Text

Description automatically generated

**init():** The init() method is designed only to called only once in a life time of servlet. This init() method is called when a servlet is first created. The servlet is first in the state of doesnot exist. When the servlet is created immediately the init() method is invoked immediately and it goes to the initialized state. The init() method is used to one time initialization things. It is used to initialize the variables.

When user clicks an url that corresponds to servlets then the servlet is created. When the servlet is created then the init() method is invoked immediately. In the init() method we write the code that corresponding to the one time initializations.

Syntax of init() method is:

public void init() throws ServletException

{

Initialization code

}

**service():** The service method is invoked when the initialization is completed. It is the main method to perform task. Whenever a client request for any task to do then the service method is used to perform those tasks and give the responses to the client requests. In order to handle and give back response to the client’s request we write the corresponding code in the service() method. Each time the server receives a request for a servlet, the server creates a new thread and calls the service()method. The service method checks the Http request type(GET or POST) and calls doGet() or doPost() methods.

Syntax:

public void service(ServletRequest req, ServletResponse res)

{

Actual business logic for providing service to client

}

**destroy():** The destroy() method is also called only one time in the entire life cycle of the servlet. It is called at the end of the life cycle of servlet. This destroy() method will delete the object that has been created for the client’s request. It gives our servlet a chance to close database connections and perform other clean-up activities. After the destroy() method is called,the servlet object is marked for garbage collection.

Syntax:

public void destroy()

{

Closing database connections

}