Client-Server Architecture and Scripting
Client/Server Architecture

• Two-tier system
  • Consists of a client and a server

• Server or back end
  • Is where the files are kept, and usually also contains a database from which a client requests information

• Client or front end
  • Responsible for user interface
  • Gathers information from the user, submits it to a server, and then receives, formats, and presents the results returned from the server
Client/Server Architecture (continued)

Figure 1-12: The design of a two-tier client/server system
Client/Server Architecture (continued)

• The Web is built on a two-tier client/server system
  • Requests and responses through which a Web browser and Web server communicate happen with HTTP (on top of the TCP/IP protocol suite)

• Three-tier, or multitier, client/server system
  • Client tier
  • Processing tier
  • Data storage tier
Client/Server Architecture (continued)

Figure 1-13: The design of a three-tier client/server system
Client-Side Scripting

• Scripting runs on a local browser
• Turns web pages into applications such as games or calculators
• Change the contents of a Web page after a browser has rendered it
• Create visual effects such as animation, and control the Web browser window itself
Server-Side Scripting and PHP

• Server-side scripting
  • Scripting language executed from a Web server
  • Popular languages/frameworks: nodejs (server-side JavaScript), PHP, ASP.NET (supports C#, Python), and JSP (Java only)
  • Server sends HTML to the client (client never sees any of the server-side scripts)

• Reasons for using a server-side scripting language:
  • to develop interactive web sites that communicate with a database
  • security, e.g. by using server-side sessions

• Server-side scripting language cannot:
  • Access or manipulate a web browser
  • Run on a client tier
Server-Side Scripting and PHP (continued)

Figure 1-14: How a Web server processes a server-side script
Client-side vs. Server-side

• Allow client to handle user interface processing and data validation
• Have the web server perform intensive calculations and data storage
• Important to perform as much processing as possible on the client