What Is JavaScript and What Can It Do?

- Larry Ullman, in his *Modern JavaScript: Develop and Design* (Peachpit Press, 2012), has an especially succinct definition of JavaScript: it is an object-oriented, dynamically typed, scripting language.
- Although it contains the word *Java*, JavaScript and Java are vastly different programming languages with different uses. Java is a full-fledged compiled, objectoriented language, popular for its ability to run on any platform with a Java Virtual Machine installed. Conversely, JavaScript is one of the world's most popular languages, with fewer of the object-oriented features of Java, and runs directly inside the browser, without the need for the JVM. Although there are some syntactic similarities, the two languages are not interchangeable and should not be confused with one another.
- JavaScript is object oriented in that almost everything in the language is an object. For instance, variables are objects in that they have constructors, properties, and methods. Unlike more familiar object oriented languages such as Java, C#, and Visual Basic, functions in JavaScript are also objects. Given that JavaScript approaches objects far differently than other languages, and does not have a formal class mechanism nor inheritance syntax, we might say that it is a *strange* object-oriented language. JavaScript is dynamically typed (also called weakly typed) in that variables can be easily (or implicitly) converted from one data type to another.
- In a programming language such as Java, variables are statically typed, in that the data type of a variable is defined by the programmer (e.g., int abc) and enforced by the compiler. With JavaScript, the type of data a variable can hold is assigned at runtime and can change during run time as well.

Client-Side Scripting

- The idea of **client-side scripting** is an important one in web development. It refers to the client machine (i.e., the browser) running code locally rather than relying on the server to execute code and return the result.
- There are many client-side languages that have come into use over the past decade including Flash, VBScript, Java, and JavaScript. Some of these technologies only work in certain browsers, while others require plug-ins to function.

- Figure 6.1 illustrates how a client machine downloads and executes JavaScript code. There are many advantages of client-side scripting:
  - Processing can be offloaded from the server to client machines, thereby reducing the load on the server.
  - The browser can respond more rapidly to user events than a request to a remote server ever could, which improves the user experience.
  - JavaScript can interact with the downloaded HTML in a way that the server cannot, creating a user experience more like desktop software than simpleHTML ever could.
  - The disadvantages of client-side scripting are mostly related to how programmers use JavaScript in their applications. Some of these include:
  - There is no guarantee that the client has JavaScript enabled, meaning any required functionality must be housed on the server, despite the possibility that it could be offloaded.
  - The idiosyncrasies between various browsers and operating systems make it difficult to test for all potential client configurations. What works in one browser, may generate an error in another.
  - JavaScript-heavy web applications can be complicated to debug and maintain. JavaScript has often been used through inline HTML hooks that are embedded into the HTML of a web page.

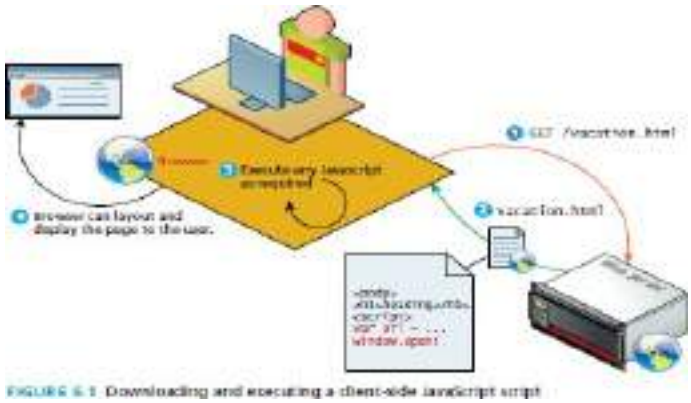
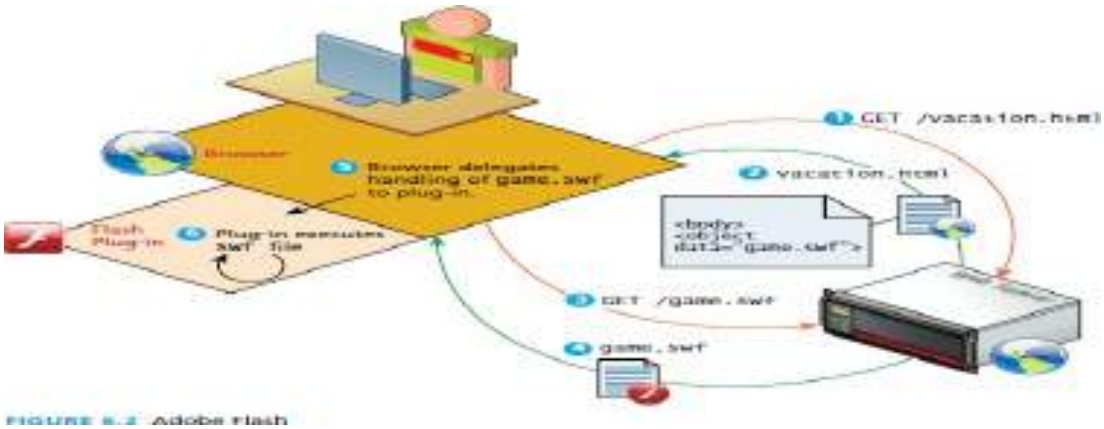


FIGURE 6.1 Downloading and executing a client-side JavaScript script

- There are two other noteworthy client-side approaches to web programming. Perhaps the most familiar of these alternatives is **Adobe Flash**, which is a vectorbased drawing and animation program, a video file format, and a software platform that has its own JavaScript-like programming language called **ActionScript**.
- Flash is often used for animated advertisements and online games, and can also be used to construct web interfaces. It is worth understanding how Flash works in the

browser. Flash objects (not videos) are in a format called SWF (Shockwave Flash) and are included within an HTML document via the <object> tag. The SWF file is then downloaded by the browser and then the browser delegates control to a plug-in to execute the Flash file, as shown in Figure 6.2.

- A **browser plug-in** is a software add-on that extends the functionality and capabilities of the browser by allowing it to view and process different types of web content. It should be noted that a browser plug-in is different than a **browser extension**— these also extend the functionality of a browser but are not used to process downloaded content. For instance, FireBug in the Firefox browser provides a wide range of tools that help the developer understand what's in a page; it doesn't really alter how the browser displays a page.



- The second (and oldest) of these alternatives to JavaScript is **Java applets**. An**applet** is a term that refers to a small application that performs a relatively smalltask. Java applets are written using the Java programming language and are separateobjects that are included within an HTML document via the <applet> tag,downloaded, and then passed on to a Java plug-in. This plug-in then passes on theexecution of the applet outside the browser to the Java Runtime Environment (JRE)that is installed on the client's machine.
- Figure 6.3 illustrates how Java applets workin the web environment. Both Flash plug-ins and Java applets are losing support by major players for a number of reasons. First, Java applets require the JVM be installed and up to date, which some players are not allowing for security reasons (Apple's iOS poweringiPhones and iPads supports neither Flash nor Java applets). Second, Flashand Java applets also require frequent updates, which can annoy the user andpresent security risks. With the