Introduction to Programming

OBJECTIVES: This material will introduce you to

- Programming languages
- The common elements found in most programming languages
- JavaScript
- Object based languages
- What software you will need to create and run JavaScript code

About programming

What exactly is programming? Without it, computers won't work. Who can learn how to program? Even young children, using a special programming language called Logo, are able to write programs. As an introduction to programming and programming languages, we will spend a few weeks learning about JavaScript, a programming language designed to work with Web pages.

A **program** is a set or sequence of instructions entered into a computer to perform work. Programs are written in **programming languages**, such as Visual Basic, Java, C++, or JavaScript. The term **script** is sometimes used to denote a JavaScript program. Programming languages are made up of special terms (called **keywords**), commands, and ways to represent information (data). Programs at first may seem a little strange, but they basically contain English words.

However, the computer cannot carry out instructions written in English. It only understands instructions written in its very own language, **machine code,** or **binary code**. Binary code consists of patterns made up of nothing but 1s and 0s. Can you imagine writing instructions for a computer using just 0s and 1s? Actually, the first programmers had to do this. But then computer scientists figured out a better way. They devised a mechanism to allow programmers to write instructions similar to human language, and then used the computer itself to translate those instructions into machine code.

How does a program get translated into binary code? Well, another program does the job! There are two techniques used to translate a program into binary code. One is called a **compiler**. A compiler looks at all the instructions in the program, and then translates the entire list into a full set of binary instructions. The second type is called an **interpreter**. An interpreter looks at only one instruction at a time, and then translates it into binary code, and the computer executes those instructions. The language we will study, JavaScript, is an interpreted language. When you execute a JavaScript program, your Web browser, i.e. Netscape or Internet Explorer, does the job of interpreting your JavaScript code.

Programming is writing the sequence of instructions that, after being translated into binary code, will allow the computer to complete a task.

Common elements found in most programming languages

The language we will study in CIS101, JavaScript, has elements common with many other programming languages. Most languages contain ways to represent information (data), statements that examine or transform (change) data, statements that collect (input) data, and statements that format and display (output) data.

The common elements of most languages (including JavaScript) consist of the following:

- variables
- data types
- expressions
- keywords
- statements
- functions

A variable is a reference to a storage locations inside main memory (RAM) that contains information relevant to a program. For example, if you used the computer to register for your courses, the program that processed your registration would need a variable to store the number of credits you have registered for. The information stored in variables is in binary format (i.e. only 0s and 1s). The information is also volatile. In other words, just like other information stored in memory, its value in memory can be lost if it is not saved before the machine is turned off.

A **Data Type** is a category of information that a programming language is capable of representing and processing. Although data types vary from language to language, the three found in JavaScript are **Number**, **String**, and **Boolean**. Numbers can represent either integers (whole numbers) or real numbers (numbers with a fractional portion). A String is any sequence of letters, numbers, or other printable characters enclosed with quote marks (either “” or ‘’). Boolean data can only have two possible values: true or false.

An **expression** is the computer science term for a formula that generates a value. Expressions can contain variables, arithmetic operators like plus (+) and minus (-), and other elements.

A **keyword** or **reserved word** is a word that has a special meaning in JavaScript, and the word cannot be used for any other purpose such as the name of a variable.

A **statement** is a command that can be carried out by a program, such as statements that repeat (for loop and while loop), or statements that examine a condition and branch in different directions depending on the condition (if statement).

A **function** is a part of a program that carries out a specific task or purpose. It is a kind of mini-program. A function has a name, and the function name represents the set of statements inside the function that perform its intended task. So when you use the function name in your code, the statements associated with that name are executed.

About JavaScript

Over the next few weeks you will be learning how to program using JavaScript. The goal for this course is not to make you a programmer. Instead the course's goal is to introduce you to some simple programs, and give you an appreciation of the work involved in creating a program. Since computer software is constantly changing, understanding how programs are put together will help prepare you to learn new tools.

In 1995 Netscape began to design and implement a new language intended to add interactivity to Web pages. The language was christened "LiveScript" to reflect its dynamic nature, but was quickly renamed JavaScript, a decision that has caused great confusion as to the relationship between Java and JavaScript.

Although their names make them seem related, there is no direct connection between JavaScript and Java, a complex cross-platform programming language developed by Sun Microsystems to write standalone applications or applets attached to HTML pages. JavaScript is also not a subset of Java (meaning that it is not just scaled-down Java with some of the features left out). Although JavaScript is structured in ways that are somewhat similar to Java, there are many differences between the two as well.

JavaScript is a programming language used to create dynamic Web pages. The purpose of JavaScript is to allow code downloaded along with HTML to be executed on the **client**. The client computer is the computer you are sitting in front of when you surf the Web. When you click on a hyperlink, HTML code is copied and transmitted across the Internet from another computer, called the **server**, onto your machine (the client). Your browser, following the instructions contained in HTML, then displays the page. By including JavaScript code with the HTML and allowing it to execute on the client, there is a dramatic decrease in the time consuming back and forth between the client and the server. Before the existence of JavaScript, programs that added interactivity to Web pages ran on the server, resulting in very slow response times. JavaScript is called a "lightweight" programming language because it is not compiled, but is interpreted line by line. This eliminates the need for a compiler, but requires the browser to be JavaScript enabled, which means your browser carries out the job of interpreting the JavaScript code.

You can use JavaScript to:

- Display different HTML depending on whether the browser is Netscape or Internet Explorer.
- Validate user input data prior to sending data to the server for processing
- Create dynamic effects like animation, scrolling text, swapping of images, and manipulation of layers
- Check for plug-ins being installed

Some important features of JavaScript:

- It is currently supported by both Netscape and Internet Explorer, although there are many examples of scripts that appear differently depending on which browser you are using.
- JavaScript source code is usually directly embedded in an HTML document.
- JavaScript programs are event driven. As we will learn in Lesson 05, an event is a user action that the computer can respond to. So an event driven program can respond to user events like clicking a button or moving the mouse.
- It is compact and relatively easy to learn.
- It is an object-based scripting language.
- It is an interpreted language (interpreted by the browser).

Object based languages

JavaScript is an object-based language. Other object-based languages include Visual Basic and VB Script. An **object** is a "package", a collection of **properties** (variables) and **methods** (functions) all combined under a single name. The properties of an object refer to its characteristics. The methods of an object refer to actions it is capable of executing. For example, imagine that there was an object named car. We could say that the car object possesses several properties: make, model, year, and color, for example. We might even say that car has some methods, or actions it can perform: go(), stop(), and reverse(). Although car is obviously fictional, you can see that its properties and methods all relate to a common theme. Objects are an important component of modern programming languages such as C++ and Java.

Object based languages have built-in objects already defined and available for use. In JavaScript, some examples include the document object, the location object, and the window object. You will use some of these built-in objects in your programs, beginning in Lesson 01 using the document object to display text. In JavaScript, the set of pre-existing objects is known as the "Document Object Model", or DOM.

The DOM is a hierarchy of objects "built in" to JavaScript. Most of these objects are directly related to characteristics of the Web page or browser. DOM defines the properties and methods for each object. When you write JavaScript code, all the objects in DOM are available for use.

Software you will need to Write JavaScript Code in CIS101

Since JavaScript is embedded in HTML code, you will continue to use 1stPage 2000 to write your JavaScripts. For instructions on how to get started using 1st Page 2000, see Appendix B, "Using 1st Page 2000 to Write JavaScripts."

The screen shots and code for this book use the browser Internet Explorer 6.0. Since there are substantial differences between how Netscape and Internet Explorer execute JavaScript, your output might appear quite different from the way it is presented in this book. If you are using Netscape and the output does not seem right, try it with Internet Explorer.

Key Terms and Definitions

- **program –** a sequence of instructions entered into a computer to perform work
- **programming language –** a set of rules, syntax requirements, and data representation techniques that can be used to create a program, or a set of instructions for a computer.
- **keywords** – words that are part of a computer language that have been given a specific and precise meaning, allowing the language to generate programs.
- **binary code** – also known as machine code, it is the series of 0s and 1s that represent information and ocmputer instructions. Computers can only process information and instructions that have been translated into binary format.
- **compiler** – a program that produces binary code (machine code) from instructions written in another programming language. A compiler first examines all the instructions in the input code file, then creates a separate file containing all the machine code needed to execute the program.
- **interpreter** – an interpreter also translates instructions from a programming language into machine code. However, it performs this translation one line at a time. JavaScript is an example of an interpreted language. Your Web browser carries out the task of interpreting your JavaScript code.
- **variable** – named location in the computer's main memory that stores information in binary format (0s and 1s).
- **data type** – a category of information that a programming languages is capable of representing and processing. The primary data types in JavaScript are Number, representing numeric data, String, representing character data, and Boolean, representing logical data.
- **expression** – a formula that is part of a program that calculates a value.
- **statement** – command that computer can carry out.
- **function** – mini program that carries out a specific task. Also called methods when referring to objects.
- **client** – computer that requests copies of HTML and JavaScript code for display.
- **server** – computer that hosts Web pages that may contain JavaScripts.
- **object** – package that is part of a programming language that combines data (properties) and functions or methods that act on the data.
- **properties** - data that is part of an object.
- **methods** – functions or actions an object can carry out or complete.
- **Document Object Model** – hierarchy of built in pre-defined objects available in JavaScript.