Tutorial 1: *Programming with JavaScript*

Strings

String Constants

A **string constant** is a sequence of 0 or more characters that is treated as a single item and is enclosed in quotation marks.

String constants can be assigned to variables, displayed in picture boxes, and combined by an operation called concatenation (denoted by +).

String constant examples:
- "abc"
- "Hello"
- "123 First Street"
- "555-22-2323"
- "(847) 555-1212"

Arithmetic should never be performed on strings. For example "2" + "3" is an invalid statement.

String Expressions

Two strings can be combined to form a new string consisting of the strings joined together. The joining operation is called **concatenation** and is represented by a plus sign (+).

For instance, "good" + "bye" is "goodbye"

A combination of strings and plus signs that can be evaluated to form a string is called a ***string expression***.

String expression examples:
- "abc" + "def"
- "123" + "456"
- "U" + "S" + "A"

String Variable Declaration

A **string variable** is a named memory location that can only be assigned *string constants* or *string expressions*.

The **var** statement is said to **declare** a variable. Declaring variables at the beginning of each procedure is regarded as good programming practice because if makes programs easier to read and helps prevent certain types of errors.

String Variable Declaration *(continued from previous page)*

There are six data types in JavaScript:

- Integers
- Floating point (numbers with decimals)
- Boolean values (true or false)
- String values (may be enclosed by single ' or double " quotes)
- Arrays of any of the above types
- Objects

There is also a NULL value which can be any of the types above, exclusive of arrays and objects.

The type of the variable is determined at its creation time, by the type of variable assigned to it. Therefore the example:

```
var  firstvar = 3;
```

creates an integer variable. If the variable value is later changed to:

```
firstvar = "Three" ;
```

it becomes a string. Note here that when the variable is created, the keyword "var" is used to create it. When referencing the variable, after its creation, or changing its value, the keyword "var" is not used.

Here is an example of an explicit *string* data type *variable* declaration using *typecasting*. This forces the variable named x to be the data type of string even though no value has been assigned to it.

```
var x = new String();
```