PRIMITIVE DATA TYPES IN HTML

4.4 Primitive Types

- Five primitive types
  - Number
  - String
  - Boolean
  - Undefined
  - Null
- There are five classes corresponding to the five primitive types
  - Wrapper objects for primitive values
  - Place for methods and properties relevant to the primitive types
  - Primitive values are *coerced* to the wrapper class as necessary, and vice-versa

4.4 Primitive and Object Storage

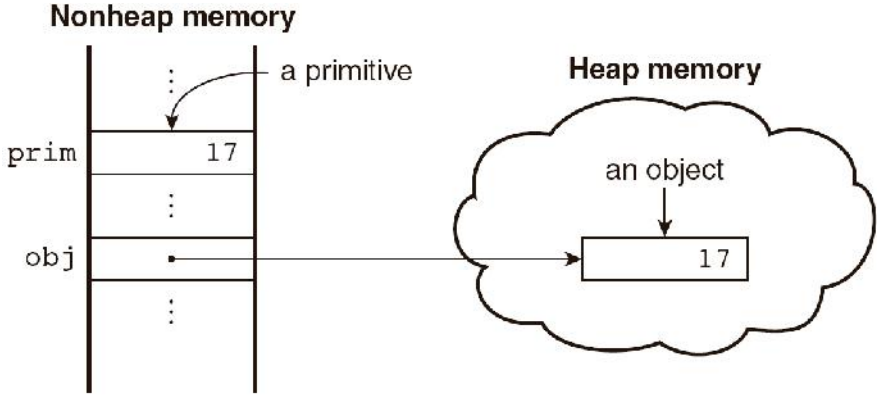

Figure 4.1 Primitives and objects

4.4 Numeric and String Literals

- Number values are represented internally as double-precision floating-point values
  - Number literals can be either integer or float
  - Float values may have a decimal and/or and exponent

- A String literal is delimited by either single or double quotes
  - There is no difference between single and double quotes
  - Certain characters may be *escaped* in strings
    - \' or \" to use a quote in a string delimited by the same quotes
    - \\ to use a literal backspace
  - The empty string '' or "" has no characters

4.4 Other Primitive Types

- Null
  - A single value, null
  - null is a reserved word
  - A variable that is used but has not been declared nor been assigned a value has a null value
  - Using a null value usually causes an error
- Undefined
  - A single value, undefined
  - However, undefined is not, itself, a reserved word
  - The value of a variable that is declared but not assigned a value
- Boolean
  - Two values: true and false

4.4 Declaring Variables

- JavaScript is *dynamically typed*, that is, variables do not have declared types
  - A variable can hold different types of values at different times during program execution
- A variable is declared using the keyword var

```
var counter,
index,
pi = 3.14159265,
quarterback = "Elway",
stop_flag = true;
```

4.4 Numeric Operators

- Standard arithmetic
  - + * - / %
- Increment and decrement
  - -- ++
  - Increment and decrement differ in effect when used before and after a variable
  - Assume that a has the value 3, initially
  - (++a) * 3 has the value 24
  - (a++) * 3 has the value 27

a has the final value 8 in either case

4.4 Precedence of Operators

Operators	Associativity
++, --, unary -	**Right**
***, /, %**	**Left**
+, -	**Left**
>, <, >= ,<=	**Left**
==, !=	**Left**
===,!==	**Left**
&&	**Left**
||	**Left**
=, +=, -=, *=, /=, &&=, ||=, %=	**Right**

4.4 Example of Precedence

```
var a = 2,
b = 4,
c,
d;
c = 3 + a * b;
// * is first, so c is now 11 (not 24)
d = b / a / 2;
// / associates left, so d is now 1 (not 4)
```

4.4 The Math Object

- Provides a collection of properties and methods useful for Number values
- This includes the trigonometric functions such as sin and cos
- When used, the methods must be qualified, as in Math.sin(x)

4.4 The Number Object

- Properties
  - MAX_VALUE
  - MIN_VALUE
  - NaN
  - POSITIVE_INFINITY
  - NEGATIVE_INFINITY
  - PI
- Operations resulting in errors return NaN
  - Use isNaN(a) to test if a is NaN
- toString method converts a number to string