Objects and Arrays

Properties

- A lot of JavaScript values have other values associated with them
- These associations are called **properties**
- Every string has a property called length, which refers to a number, the amount of characters in that string
- Properties can be accessed in two ways:

```
var text = "purple haze";
console.log(text["length"]);
console.log(text.length);
```

- The properties of a string value can not be changed!

Objects

- Objects have their own set of members in the form of properties
- You are free to modify these, remove them, or add new ones
- E.g.:

```
var cat = {colour: "grey", name: "Spot",
           size: 46};
cat.size = 47;
console.log(cat.size);
delete cat.size;
console.log(cat.size);
console.log(cat);
```

- The keyword delete cuts off properties
- Trying to read a non-existent property gives the value undefined

Objects

- An object is a referenceable container of name/value pairs
  - usually implemented as a hash-table
- The names are strings
  - or other elements such as numbers that are converted to strings
- The values can be any of the data types, including other objects
- New members can be added to any object at any time by assignment

- Arrays and functions are implemented as special kinds of objects

Accessing Properties

- The dot notation can be used when the subscript is a string constant in the form of a legal identifier
- Because of an error in the language definition, reserved words cannot be used in the dot notation, but they can be used in the subscript notation

```
var thing = {"gabba gabba": "hey"};
console.log(thing["gabba gabba"]);

thing["if"] = "funny";
console.log(thing["if"]);
```

In operator

- The operator **in** can be used to test whether an object has a certain property

```
var set = {"Spot": true};
// Add "White Fang" to the set
set["White Fang"] = true;
console.log("Spot" in set);
// Prints true
```

Date Objects

- JavaScript has a build in Date object
  - The Date object can store a time of day as well as a date
  - And has a constructor for creating new date objects

```
var when = new Date(1980, 1, 1);
console.log(when);
```

- The Date constructor can be used in different ways:

```
console.log(new Date());
console.log(new Date(1980, 1, 1));
console.log(new Date(2007, 2, 30, 8, 20, 30));
```

- The month numbers go from 0 to 11,
  - which can be confusing
  - Especially since day numbers *do* start from 1

Date objects has a number of get... methods

- The content of a Date object can be inspected with a number of get... methods:

```
var today = new Date();
console.log("Year: ", today.getFullYear(), ", month: ",
       today.getMonth(), ", day: ", today.getDate());
console.log("Hour: ", today.getHours(), ", minutes: ",
       today.getMinutes(), ", seconds: ", today.getSeconds());
console.log("Day of week: ", today.getDay());
```

ARRAYS

Arrays in JavaScript

- Are implemented as hashtable objects
  - This makes them very well suited to sparse array applications
- The values are located by a key, not by an offset
  - This makes JavaScript arrays very convenient to use, but not well suited for applications in numerical analysis (to slow)
- When you construct an array, you do not need to declare a size
  - Arrays grow automatically, much like a C# collection
- There are two ways to make a new empty array:

```
var myArray = [];
var myArray = new Array();
```

Array Initializing

- Arrays have a literal notation, similar to that for objects:

```
myList = ['oats', 'peas', 'beans', 'barley'];

monthLengths = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

slides = [
    { url: 'slide0001.html', title: 'Looking Ahead' },
    { url: 'slide0008.html', title: 'Forecast' },
    { url: 'slide0021.html', title: 'Summary' }
];

console.log('Number of days in february: ' + monthLengths[1]);
```

Arrays Are Not Typed

- They can contain numbers, strings, booleans, objects, functions, and arrays
- You can mix strings and numbers and objects in the same array
- The first index in an array is usually zero

length

- When a new item is added to an array and the subscript is an integer that is larger than the current value of length, then the length is changed to the subscript plus one
  - This is a convenience feature that makes it easy to use a for loop to go through the elements of an array.

```
myList = ['oats', 'peas', 'beans', 'barley'];

console.log('The length of myList is ' + myList.length);

myList[27] = 'banana';

console.log('The length of myList is ' + myList.length);
```

Arrays Specialities

- The method push can be used to add values to it
- The method pop takes off and returns the last value in the array
- Join builds a single big string from an array of strings
  - The parameter it is given is pasted between the values in the array

```
var mack = [];
mack.push("Mack");
mack.push("the");
mack.push("Knife");
console.log(mack.join(" "));
console.log(mack.pop());
console.log(mack);
```