

JavaScript Functions

CSE 264

J. Femister

Defining and Invoking

```
function distance(x1, y1, x2, y2) {  
    var dx = x2 - x1;  
    var dy = y2 - y1;  
    return Math.sqrt(dx*dx + dy*dy);  
}
```

Nested

```
function hypotenuse(a, b) {  
  function square(x) { return x*x; }  
  return Math.sqrt(square(a) + square(b));  
}
```

Function Literals

```
function f(x) { return x*x; } // function statement  
var f = function(x) { return x*x; }; // function literal
```

```
f[0] = function(x) { return x*x; }; // Define a function and store it  
a.sort(function(a,b){return a-b;}); // Define a function; pass it to  
// another
```

```
var tensquared = (function(x) {return x*x;})(10); // Define and invoke
```

Function Constructor

```
var f = new Function("x", "y", "return x*y;");
```

```
function f(x, y) { return x*y; }
```

Optional Arguments

```
function copyPropertyNamesToArray(o, /* optional */ a)
{
    if (!a) a = []; // If undefined or null, use a blank array

    for(var property in o)
        a.push(property);
    return a;
}

a = a || [];
```

Varargs Functions/Arguments Object

```
function max(/* ... */) {  
    var m = Number.NEGATIVE_INFINITY;  
    for(var i = 0; i < arguments.length; i++)  
        if (arguments[i] > m)  
            m = arguments[i];  
    return m;  
}
```

```
var largest = max(1, 10, 100, 2, 3, 1000, 4, 5, 10000, 6);
```

Functions as Data

```
function square(x) { return x*x; }  
var a = square(4); // a contains the number 16  
var b = square; // Now b refers to the same function that  
square does  
var c = b(5); // c contains the number 25  
var o = new Object;  
o.square = function(x) { return x*x; } // function literal  
y = o.square(16); // y equals 256  
var a = new Array(3);  
a[0] = function(x) { return x*x; }  
a[1] = 20;  
a[2] = a[0](a[1]);
```


Function Properties

```
uniqueInteger.counter = 0;
```

```
function uniqueInteger() {  
    // Increment and return our "static" variable  
    return uniqueInteger.counter++;  
}
```