Announcements

Project 2a due Monday
- Kasparov and Deep Jr. are tied

Functions & Abstraction

A function is a package for an algorithm; once written, it can be used over and over.

The Package

Functions have a specific syntax

```
function <name> ( <parameter list> ) {
  <function definition>
}
```
- `<name>` names are identifiers; start w/letter
- `<parameter list>` is the input variables, a list separated by commas
- `<function definition>` is just the program to do the work

A Sample Function

Compute the Body Mass Index when the inputs are in metric

```
function <name> ( <parameter list> ) {
  <function definition>
}
```

```
// Figure Body Mass Index in metric units
return weight/kg / (heightM * heightM);
```

Writing Functions

Most programming is done by writing functions, so learning the form is key

```
function bmi ( weight, height ) {
  // Figure Body Mass Index in English units
  var heightFt = height / 12; // Change to feet
  return 703 * weightLBS / (heightFt * heightFt);
}
```

Declarations

A function is declared by writing down the “package”… the function is used when it is called

```
// Compute BMI in either metric or English
if (units == “English”) return bmi(weight, height);
else return bmiM(weight, height);
```
Summarizing

Declaration: the function "package," says what happens when the function runs.

Call: the function use, asks for the computation to be run:

- There is only one function declaration.
- There can be many calls ... functions are reusable.
- In JS, functions tend to be grouped together but the calls go where they are needed.

Gold Function

Suppose we compute "weight in Au"

Worth = \( (\text{Weight} \times 12) \times 368.4 \)

Recall Testing Template

No one writes perfect programs the first time ... smart programmers check.

To test, have a standard page handy:

```html
<body>
<script language="JavaScript">
  // Put your JavaScript code here
</script>
</body>
```
Declare the Function

Put a function declaration in <script>

```
function worthless ( weight ) { 
  // Compute the dollar value 
  // at weight of $588 40/tonne or 
  return weight * 12 * 588.40; 
}
```

Try The Function

Questionably, the best practice is to test everything

Function Features

Reviewing properties of functions

- Selecting names ... don't use alert()
- Parameter variables ... don't need to be declared, but work like local variables

```
var weight = weight12; // Figure BMI in English (height in ft)
    height = height12; // Change to feet
    return 4.89 + weight12 / (height12 * height12);
```

Function Features (cont.)

- Scope of Reference ... refers to where in the program a variable is "known," i.e., where its value can be referenced and/or assigned

```
function bmi ( units, height, weight ) { 
  // Compute BMI 
  if ( units == "English")
    return bmi(weight, height);
  else
    return bmi(weight, height);
}
```

Function Features (cont.)

- Global ... declared outside of functions

```
var priceless = 368.40;
```

Function Features (cont.)

- Parameters vs Arguments ... parameters are the “formal” variables in the function declaration; arguments are the same thing in the call

```
index = bmi("English",12,200)
```
Summary

Functions are packages for algorithms
- They follow a series of rules, that quickly become intuitive
- Functions have both a declaration and a call
- Functions have both parameters (in the declaration) and arguments (in the call)
- Scope refers to the region of a program where a variable is “known”

functions show what it means to be “precise” enough when describing what to do