Announcements

* The Project 4 due Friday, 2/19. Note: this project includes the week 4 and 5 lab assignments.
* The midterm is being scanned. I’m waiting for results to be sent to me.
* The current version of they syllabus is now linked in the Syllabus Description on Canvas (even though the link and text are for an older version. It’s a Canvas bug.)

Review

* Look at last week’s code: zoo and university objects
* RegExp – Regular expression object. Used for pattern matching in strings
  + Defined with forward slashes: var pattern1 = /matchThis/;
  + Or by using the *new* operator: var pattern2 = new RegExp(“/matchThisToo/”);
* Use the test method to check for a match: var foundMatch = pattern1.test(“matchThis”);  
  Passing any string to the test method that contains the string in the RegExp object will return true.

Regular Expressions

* The real power is in finding partial matches.   
  You might have used \* and ? as a wildcards in a search before.   
  With RegExp, the syntax is different, you use . to match any single character.   
  Adding a \* means to match zero to many occurrences of any character.   
  Adding a + will match one to many occurrences.
  + This pattern will match a string that contains “this”: var pattern = /this/
  + This pattern will match any string that starts with “this”: var pattern = /^(this).+/
  + This pattern will match any string ending with “this” : var pattern = /.+(this)$/

Use this code like this in the console with the above patterns: pattern.test(“trythis”);

* RegExp Properties: g – global, i – ignoreCase, m – multiline.
  + Put the property after the slash that ends the regular expression: var pattern = /this/i;
* Character groups – a group of characters that can match one in a string:  
  var pattern = /[Tt]his/; // matches capital or lower case T
* A group can be negated with a caret, ^   
  var pattern = /[^T]his/; // matches anything except a capital T
* The caret also can be used to indicate a group that matches the beginning of a string. For example, checking for capitalization of at least the first character: var pattern = /^[A-Z][a-z]\*/
* The $ specifies that a char or group must be at the end of the string. For example, now only the first char can be capitalized:   
  var pattern = /^[A-Z][a-z]\*$/
* Example – rules for naming variables: Names can contain letters, digits, underscores, and dollar signs, but names cannot begin with a digit:   
  var pattern = /^[A-Z\_$][A-Z0-9\_$]\*$/i
* Escape character – backslash is an escape character that lets you use a special character, like the dot as a dot, not for pattern matching. Example – test for a valid e-mail address:   
  (also uses {2,} to indicate a minimum of 2 characters)  
  var pattern = /^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}$/i
* Another example – check for a valid uoregon.edu address:  
  var pattern = /^[A-Z0-9.\_%+-]@uoregon.edu/i

Reference: <http://www.w3schools.com/jsref/jsref_obj_regexp.asp>