Announcements

The project 5 instructions are available on Canvas. Project 5 is due March 4th.

JSON

* JavaScript Object Notation is a protocol for transmitting and storing data (information) in the form of JavaScript objects.
* Example – Invoice for an order of honey:
* Reference: <http://www.w3schools.com/json>

------------------------------------------- We got this far on Wednesday-------------------------------------------------

The rest is for Friday… and beyond

JQuery Mobile

JQuery Mobile is a library written using jQuery that provides objects and methods for developing responsive web apps (web apps that adapt to different screen sizes- especially those of small mobile devices).

Reference: <http://jquerymobile.com>

Project 5

* getTriangleType: returns a string representing the triangle type (Equilateral, Isosceles, or

Scalene).

* ntheory:
  + Perfect number: <https://en.wikipedia.org/wiki/Perfect_number>
  + Abundant number: <https://en.wikipedia.org/wiki/Abundant_number>  
    Note: Every multiple (beyond 1) of a perfect number is abundant – could this simplify the code solution?
  + Deficient number: <https://en.wikipedia.org/wiki/Deficient_number>
  + Prime number: <https://en.wikipedia.org/wiki/Prime_number>
* Dice-namic: We’ll write the project 4 code together in class. It was part of project 4 for the other section of CIS 111.

**Appendix**

Project 4, Dice.js problem

11/p4/Dice.js.

This is an exercise in Command-Line JavaScript.

Review pp. 139-140 of JSNN.

Download Dice.js from <https://gist.github.com/mh108/9e824e7c5dca87fadcc2> to your 111/p4/folder.

Modify Dice.js as follows:

A) Complete the rollDice method. It should roll two dice and

return a number in the range [2..12].

Dice.rollDice() => 7

Dice.rollDice() => 3

B) Complete the rollDoubles method:

Dice.rollDoubles(4) => 8

Dice.rollDoubles(4) => 2

Dice.rollDoubles(6) => 3

C) Complete the roll\_number function.

Note how snake-eyes is much harder to hit than a seven:

roll\_number(7) => 4

roll\_number(7) => 2

roll\_number(2) => 67

JavaScript Object Constructors

function person(first, last, age, eye) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
}  
var myFather = new person("John", "Doe", 50  
var myMother = new person("Sally", "Rally", 48

Reference: <http://www.w3schools.com/js/js_object_definition.asp>

Fluent Interface (aka “chaining”)

<https://en.wikipedia.org/wiki/Fluent_interface>

Recommended Editor Packages

<https://atom.io/packages/atom-beautify>

<https://packagecontrol.io/packages/HTML-CSS-JS%20Prettify>