1. *Bookmark References*

Bookmark the following sites for future reference:

[Groovy homepage](http://groovy.codehaus.org)

[Groovy API](http://groovy.codehaus.org/gapi/)

[Groovy JDK](http://groovy.codehaus.org/groovy-jdk/)

[Java SE6 API](http://docs.oracle.com/javase/6/docs/api/)

2. *Number Data Types*

a. What data type is the number 2? How about 20? 200? Keep adding zeros and watch the data type change until it reaches `BigInteger`. Then do the same for 2.0.

b. Declare a variable `x` of type `def` and assign it the sum of 1 and 1.5. What is the resulting data type?

c. What do you get when you divide 5 by 2? What is the resulting data type? If you wanted to do integer division (no remainder), what method would you call?

3. *Wrapper Classes*

From the associated wrapper classes, find the min and max values for the Java primitives: `byte`, `short`, `int`, `long`, `float`, `double`.

4. *2s Complement*

Create a `byte` variable with its maximum value. What do you get when you add 1 to it?

3. *Strings and GroovyStrings*

a. How many characters are in the string "Hello, Groovy!"?

b. Define a string variable containing a name. Print a hello statement with your name using string concatenation, then using a Groovy string.

c. Demonstrate that "racecar" is a palindrome by comparing it to its reverse. Do the same with "Bob", removing case sensitivity first.

d. Define a string variable containing the sentence, "Hello, World. How are you?". Split the sentence into an array using the `split` method. Count the number of words. Do the same using the `tokenize` method.

e. Using the same sentence, use array notation (square brackets) to print the substring "World".

f. Use array notation to print the last word, but reversed.

4. *Prime Numbers*

Write a method called `isPrime` that takes an integer argument and returns a boolean. Determine whether the number is prime by dividing it by all numbers from 2 up to one less than the number.

That limit is too high, of course. How high do you have to check to be sure whether you've gone far enough?