S.J. & JESSIE E. QUINNEY COLLEGE of NATURAL RESOURCES
UtahStateUniversity

Efficient Programming Tips

1. Programming in JavaScript (or any other programming language) is very linear and ***extremely*** literal.
2. A student told me that programming is like giving instructions to a 2-year old - a perfect metaphor. In order for that 2-yr old to know you want it to pick up the toy, it first needs to know **a)** what a toy is, and **b)** if it knows what a toy is, then which toy to pick up.
   a. If the kid understands the phrase "pick up", but not "toy", it will either pick up whatever it can see first, or probably just stare at you and do nothing.
3. On the '***extremely*** literal' part, the small details matter. Programming languages are often case sensitive. They also REALLY care about things like commas, periods, parentheses, etc.
4. Take the following examples in JavaScript in GEE:

Command	Result	Why
print hello	SyntaxError: Unexpected token (1:7)	A command requires '()'s to surround what you're sending to it.
print(Hello);	Line 1: Hello is not defined	You never told it what **Hello** was. It assumes that **Hello** is a variable that you have not defined yet.
print('Hello');	Hello	The apostrophes (or quotes) define Hello as a string and therefore it knows what Hello is.
var Hello = 'Hello'; print(Hello);	Hello	'Hello' is identified as the variable Hello, therefore print knows what it is.
print(Hello); **var** Hello = 'Hello';	undefined	Remember the "linear" part?? JavaScript knows that you've defined Hello, but it needs to be defined before the print command
var Hello = 'Hello'; print('Hello');	Hello	This works, but it's wasteful since there's no need to identify the variable that = Hello

5. Try to make your scripts as concise as possible. Don't write a script that executes multiple, unrelated commands.
   a. In the lab exercises, I often provide multiple scripts. Don't copy and paste the entirety of each script into one script. That can lead to frustration, such as:
      i. You will have multiple copies of the same command. There's no need for this, you just need one of the commands. Also, I often use the same command in all scripts, but I may assign that command to a different variable name. E.G. **var** myCollection = <some command> and **var** theCollection = <same command>. Here I have the same command, but different variable names. This can lead to confusion and therefore errors (remember the 2yr-old). When you merge two scripts, make sure you don't have this situation. If you do,

R. Douglas Ramsey
Dept. of Wildland Resources
Quinney College of Natural Resources
Utah State University
Logan, Utah 84322-5230
doug.ramsey@usu.edu

then pick one version and delete the rest.

ii. A more serious case of this, and one that will certainly cause errors is when you use the same variable name, but assigned to two different commands. In this case, the second instance of this command will take precedence. In some cases, GEE will provide a warning, but not in all cases.

E.G. **var** Hello = 'Hello'

Or

var Hello = 'Goodbye'

6. So, use unique variable names.
7. Keep your code concise
8. Compartmentalize and organize your code into logical sections. I like to use section breaks like:

```
//*****************************FUNCTIONS *******************************
.
                    Put all of my functions here and at the top of the script
.
//************************************************************************
//***************************** VARIABLES ********************************
.
      If I have any standard variables like dates, etc. that I will use multiple times, I put them here
.
//************************************************************************
//***************************** LOAD DATA ********************************
.
     This is where I identify the images, image collections, and/or feature collections that I will use
.
//************************************************************************
```

This helps me keep everything in a logical order and broken up into sections with clear breaks (like the '/*'s) so I know where everything is and I can debug faster.

Bottom line, it's just like your mother always told you:

"clean your room and put things where they belong!!"

9. Learning to code can be difficult and confusing to most. Just remember to start with simple, short, and concise scripts and then progress to more difficult things.
10. *More to come……Suggestions welcome…..*

R. Douglas Ramsey
Dept. of Wildland Resources
Quinney College of Natural Resources
Utah State University
Logan, Utah 84322-5230
doug.ramsey@usu.edu

R. Douglas Ramsey
Dept. of Wildland Resources
Quinney College of Natural Resources
Utah State University
Logan, Utah 84322-5230
doug.ramsey@usu.edu