

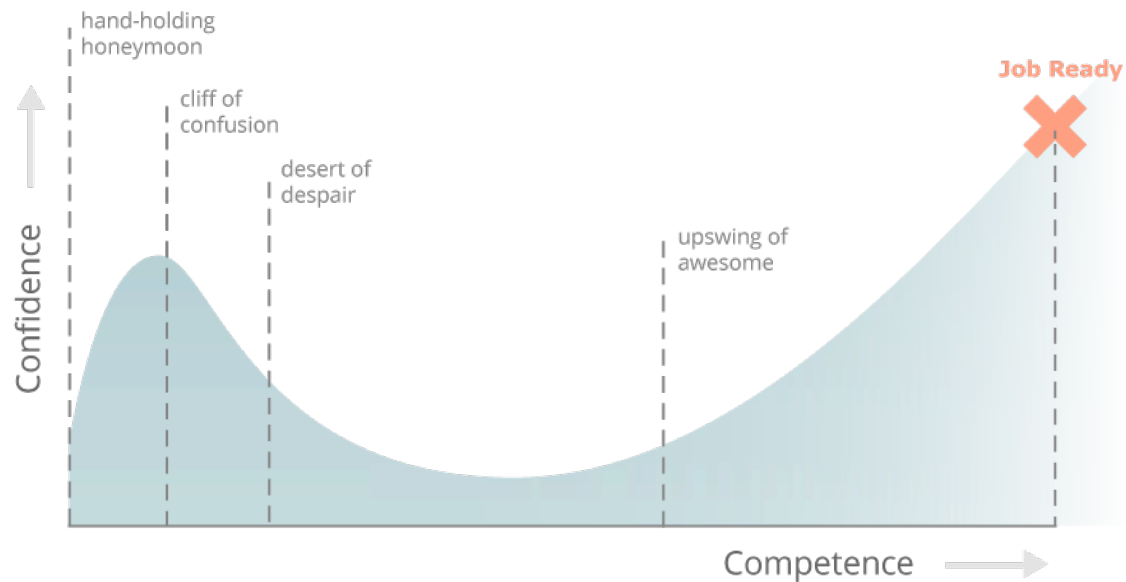
Sitting around a lunch table...



<https://flic.kr/p/4Y9Eg3>

Viking Code School

Coding Confidence vs Competence



Viking Code School Blog, "Why Learning to Code Is So Damn Hard." 2015 Feb 24.
<http://www.vikingcodeschool.com/posts/why-learning-to-code-is-so-damn-hard>



Goals of the workshop

“Why am I taking this again...?”

Hands-on

- + How to break down problems and use code to solve them.
- + Familiarity with a variety of languages and the strengths and potential of each one.
- + How APIs fit into the picture.
- + How to decide what language to apply to solve specific problems.

Applied

- + Making the most out of tools like Drupal, LibGuides, and MarcEdit
- + What skills are important to put on an RFP or job description.
- + Reasonable limits – when to call a pro!
- + How to make a worthwhile IT call.



<https://flic.kr/p/qN5Lvc>



“How do you get to Carnegie Hall?”

Practice, practice, practice



Jargon, jargon, jargon

Some of it is important!



Warm-up exercise!

- + What terms do you know?
- + What terms have you never heard of?
- + What terms has someone told you that you should have heard at a meeting, even though it seemed vaguely irrelevant at the time (and maybe still does? 😊)
- + Which term is a magic bullet for all library-wide problems? (Big hint: the answer probably isn't on the sheet.)



Section 1

Let's get started!

Think like your computer, part 1: Breaking down real-world problems

+ **Excellent resource:**

Yelton, Andromeda (2015). "Coding for Librarians: Learning by Example." *Library Technology Reports*, 51:3. <http://dx.doi.org/10.5860/ltr.51n3>

The EAD problem

- + “[a librarian] needed to export metadata: in his case, EAD files from ArchivesSpace. His short Python script finds all the resource IDs that match a given criterion, gets their EAD, and writes it to a specified destination.” (Chapter 2)

Let's go to the code (Python)!

+ <https://gist.github.com/helrond/1ef5b5bd47b47bd52f02>



<https://flic.kr/p/qN5Lvc>




“How do you get to Carnegie Hall?”

Practice, practice, practice



Let's break this problem
down

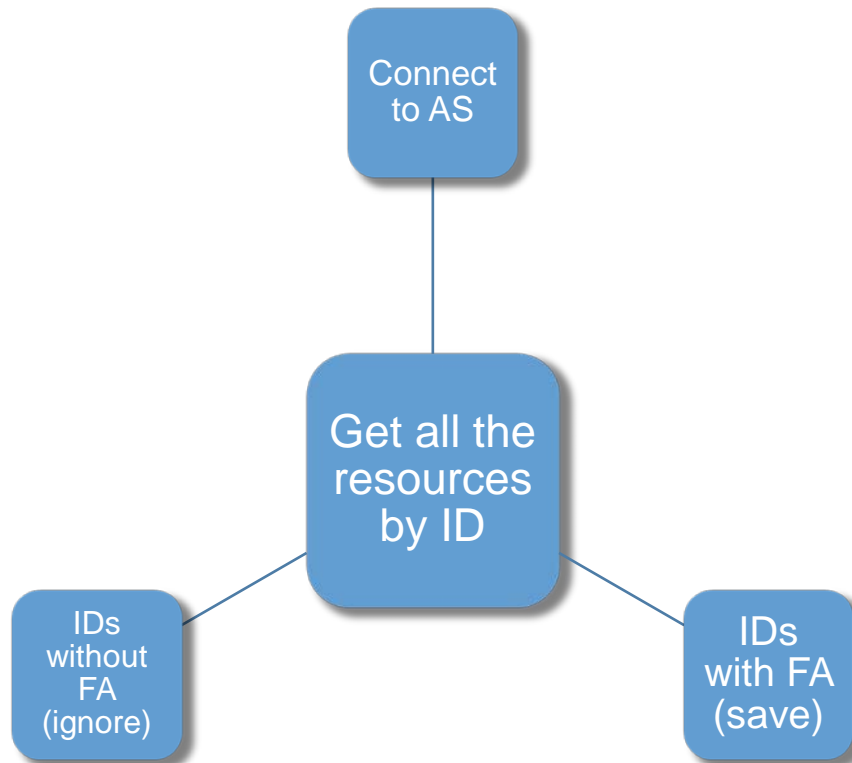




What does your script need to know to solve this problem?

- + Where is the content that I need? In some software program called ArchivesSpace.
- + Wait, so what content do I need again? EAD files.
- + What's an EAD file? It's a special kind of XML.
- + Okay.... Um... this is embarrassing. What's XML? eXtensible Markup Language.
- + (thinking) Aha! That means I'll need to be good at capturing strings, maybe not so much numbers.
- + Great! Does ArchivesSpace just have EAD files? Nope.
- + So how will I know I have an EAD file? They are the resources whose IDs have 'FA' in them.
- + What do I do when I find the stuff with the 'FA'? Save the file you find.

Diagram



Pseudocode

Connect to ArchivesSpace.

// Maybe ArchivesSpace has a publicly-available list of all of the IDs. That would make life easier.

// Assume connection is successful.

Once connected:

 Get a list of everything in the system based on ID

 For (every ID that I get):

 If the ID has the **string** "FA" in it, save it locally.

 // this is the criteria we need

 If the ID doesn't have the **string** "FA" in it, keep going.

Close the connection.



Keys to solving the problem

- + The right questions to ask
 - + What information structure do you need?
 - + Where and how is it stored?
 - + What is unique about the information I need?
 - + **How will I know when I find it?**
 - + What do I need to do with it when I find it?
- + The right facets of information
 - + Unique identifiers
 - + Data types
 - + File formats and structures




<https://flic.kr/p/9J5r8M>

A whirlwind lesson on data types & (two) paradigms



Data types (primitive)

- + Numbers
 - + Integers (short, int, long)
 - + Real (float, long)
- + Characters
 - + Char
- + Logical
 - + Boolean (true, false)



Data types (composite)

- + Arrays
 - + Associative arrays
- + Strings
- + Records
- + Object
 - + Object-oriented programming

“Why are TRUE and FALSE integers?”

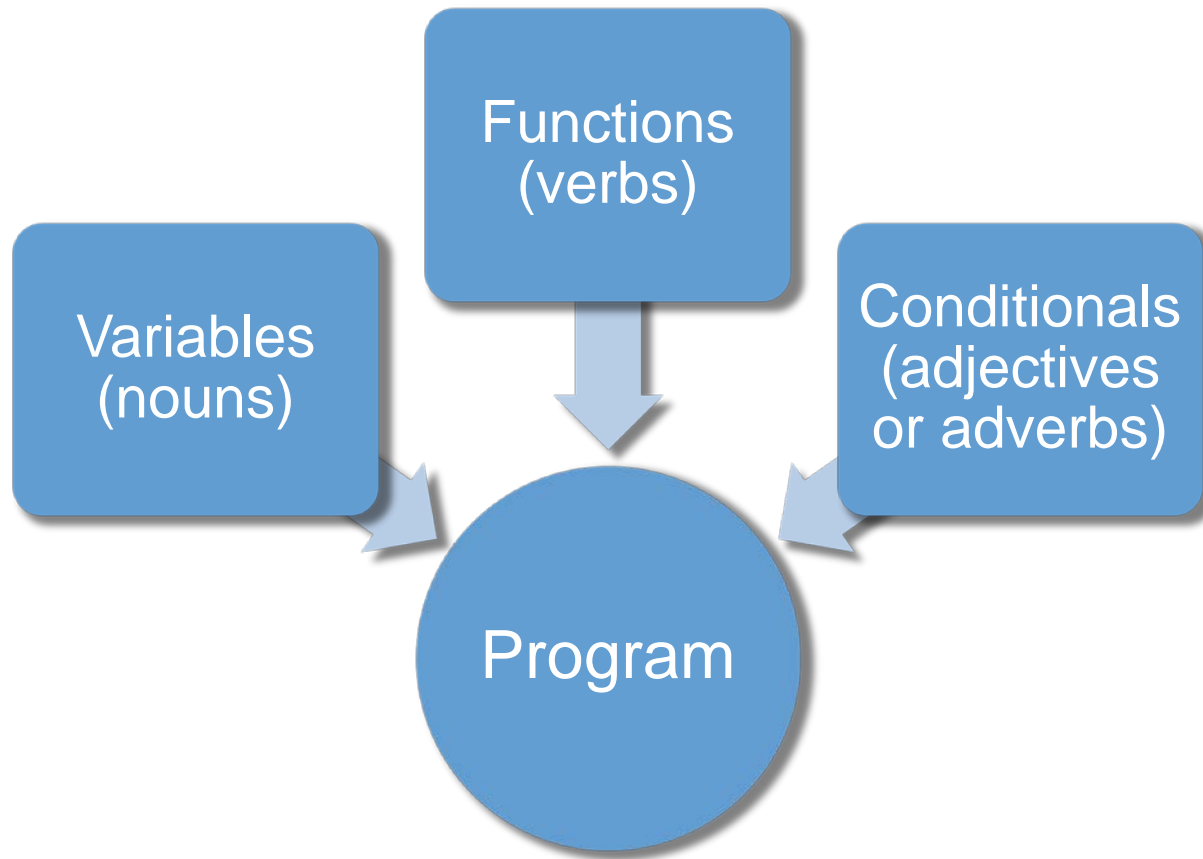
- + Lurking behind every data type is a number. Somewhere.
- + Inherently, computers... compute.
- + There's always numbers lurking behind data types.
- + Zero is the first number.
- + Associative arrays get around this a little bit.



Variable

- + A container that will hold a value
- + Can be simple or composite
- + Global variables – don't change no matter where you are in your program. Some languages have built-in global variables. Example: PHP's `$_SERVER`
- + **Object oriented programming**

Beginning to build your code



Two different programming paradigms

Functional

- + Fixed set of *things* (the types of things tend not to change)
- + Primarily adding new *operations*
- + Database query languages like SQL are considered declarative (a third animal)

Object oriented

- + Fixed set of *operations* (the functions tend not to change)
- + Primarily adding new *things*
- + Library code tends to be written in languages that are object oriented

Dicing onions

Functional

- + Define variables: knife, cutting board, bowl, vegetable = onion
- + Define functions: lift, peel, cut / dice, scrape
- + Easy to add new types of chopping methods

Object oriented

- + Define the characteristics of an onion: peel, layers, shape
- + Define methods: getting bag of onions, selecting one onion, dicing onion
- + Easy to add new types of vegetables / things to dice



Back to the EAD problem

- + Python code chosen: object-oriented programming
 - + import requests
 - + import json
 - + These are libraries of functions based on **object**
- + Variables such as string, array
- + Conditional logic / loop



<https://flic.kr/p/qN5Lvc>



Now it's your turn!

Practice, practice, practice

Exercise 1

- + A real-world problem from Code4Lib
- + Compare answers!



Section 2

The Medium Dive

Strengths of various languages

Language	Strengths	Weaknesses
Java	“Write once, run anywhere”	Slow
C	Speed	Memory management
C++	Speed	Clumsy?
PHP	Hugely popular, great documentation, web programming	Inconsistent syntax, security issues
Visual Basic	Powerful	Only runs in Windows environment
Python	Enforces best practices	None?
C#	Fast	Really only suitable for Windows
JavaScript	Only reliable way to do client-side scripting	Only useful with a web browser
Perl	Powerful and fast	Steep learning curve
Ruby	Extremely object-oriented	Slow

But what about...?

“Language”	Strengths	Weaknesses
HTML	You can't display pages on the web without it!	Too much of a mix between “look” and content
CSS	Powerful tool to separate content from layout	None ☺
SQL	Ability to query databases	Complex quickly; only useful within a database
XML	Powerful structural tool	Difficult parsing
XSLT	Designed to parse XML	Procedural and recursive
XPATH	Allows querying of XML files	Steep learning curve



Language choice vs. analysis

- + Syntax will vary
- + Available libraries* will vary
- + Your analysis will always prevail!

- + * Libraries → code that has been written for you, allowing you to stop reinventing the wheel

Common syntax + structures

- + Output / print
echo "Hello, world! Hello " . \$name . "!";
- + Comparison statements (greater than, less than, equal/not equal to)
- + Conditional statements (if/then)
if "FA" in resourceID:
 resource = resourceID["id_0"]
- + Loops
for (i = 0; i < 100; i++)
 echo "This is a FOR loop!";
- + Regular expressions
\$pattern = "^d\d\d\d-\d\d-\d\d/";
- + Combine for your code!

Example

- + “[A librarian] wrote a CoffeeScript that harvests ISBNs from a page of search results from the Blacklight discovery interface, grabs the corresponding cover image from Google, and enriches the results page with the images—all in a mere twelve lines.”
- + ~~Let's go to the code!~~



Questions

- + What information ties these systems together?
- + What does “success” look like?
- + Does it matter if I don't have a book?
- + Where does my code have to “live” for it to work?
- + Related: what is Blacklight?

Writing out the problem

When a user queries my catalog, it would be nice if the user could see book covers.

Google has book covers that I can use.

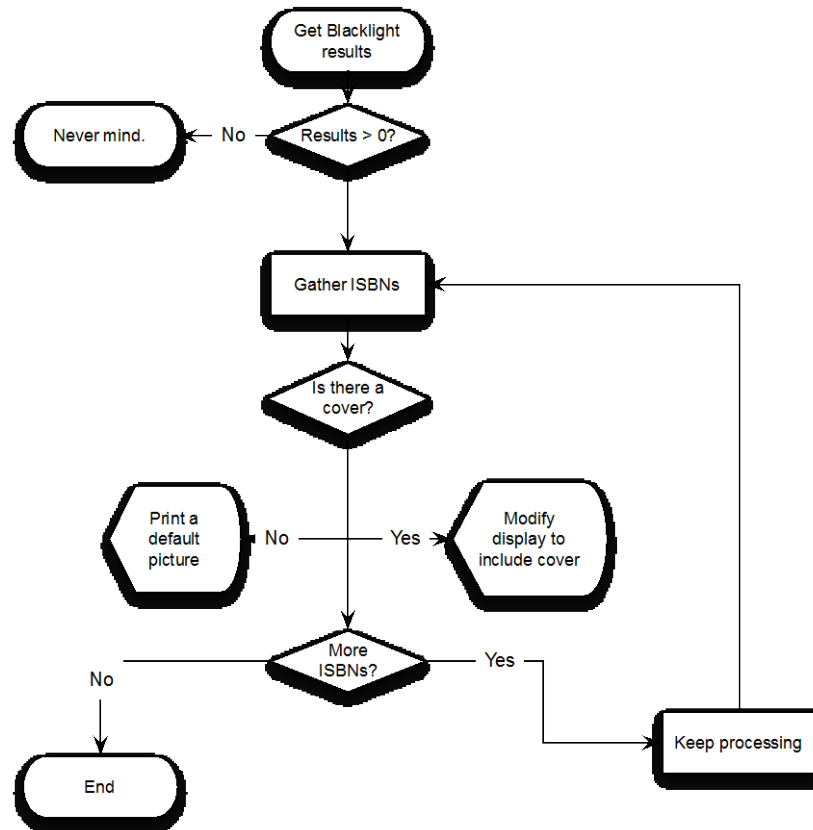
I need:

- + The ISBN of the book
- + The URL pattern for Google
- + The ability to write out an image in my catalog

Pseudocode

```
if (search results > 0), then:  
    for (each result):  
        get the ISBN;  
        go to Google's API service;  
        check if an image exists for the ISBN.  
  
        if (image exists):  
            write out HTML image tag.  
        else:  
            write out a default image.
```

Diagram





<https://flic.kr/p/qN5Lvc>



Now it's your turn!

Practice, practice, practice

Exercise 2

- + Use syntax and structure ideas to generate pseudocode to start solving the Code4Lib problem.



Section 3

The Deep Dive



<http://pixabay.com/en/road-start-beginning-intention-368719/>

Writing code

For the next hour or so, we'll be working through the example found in your workbook.

Exercise 3

- + We'll start writing code together
- + There are lots of questions in the file!
- + Compiled code and environments

- + WAMP = windows, apache, mysql, php
- + IDE = integrated development environment (Eclipse)
- + CodeAcademy (www.codecademy.com)

Exercise 4

- + Putting it all together
- + Does our code work?

Other thoughts...

- + HTML/CSS
- + Writing out report
- + Adding additional information
- + What would YOU like to see in this program?



Applying your knowledge



Getting the most out of your system

- + Content Management Systems
- + LibGuides
- + MarcEdit
- + Others
- + Know what they do and how they do it!



How to make an IT call

- + Understand your problem, as best as you can
- + Pay attention to:
 - + When the problem happens (is it predictable? timed?)
 - + How to recreate it
 - + What you've already done to solve the problem
 - + How you know the problem is solved
- + Beware and be patient:
 - + <https://xkcd.com/806/>

We'll do this exercise together...

Hi all,

I just created a new multisite instance in my production setup, and there are some weird things happening. I'm hoping you can help. Here are the issues:

- I created a new site, inscriptions.digitallibrary.vassar.edu. All went well. This is a Drupal 7 / Islandora 7 site.
- I can ingest objects perfectly fine, and they report that they are part of a parent collection, Inscriptions (inscriptions:root). E.g., see: <http://inscriptions.digitallibrary.vassar.edu/islandora/object/inscriptions:1>
- None of the objects that I ingest are listed on the parent page (<http://inscriptions.digitallibrary.vassar.edu/islandora/object/inscriptions:root>).
- Side effect: I can ingest objects into my other Drupal 7 site, einstein.digitallibrary.vassar.edu, but the same thing happens -- they don't show up in the collection that I specify as parent.
- Interestingly, when I log into my Drupal 6 site (digitallibrary.vassar.edu), I can see the "inscriptions" collection there, but not the Einstein ones.

I'm not sure what I did wrong -- I first did this in our staging area, and everything went well, once the Drupal filter was set correctly.

To add to the complexity -- when I enabled my Islandora modules, the default Audio, Book, etc., collections showed up in my D6 + D7 instances.

I'm really surprised at all of this, because things went so well in staging. None of these problems occur -- I can ingest without any issues, none of the collections seem to show up where they shouldn't, etc. I feel like the production things haven't divided up as well as they should. Complicating this, of course, is that my Einstein collection is a grant-funded one whose donors would like to see only their supported stuff show up, and I'm afraid that I'm going to delete something that is going to make that site go ka-boom.

Thanks very much for any insight you can provide!
Joanna

When to call a pro

Cost/resources

- + Do you know how to solve the problem?
 - + Do you know enough to be dangerous/security issue?
- + Have others tried to solve it?

Time

- + Do you have time to solve the problem?
- + Do you have the time to acquire the skill set to solve the problem?
- + How complicated is your issue?

Writing an RFP/RFI

- + Excellent resource: TechSoup's RFP Library
<http://www.techsoup.org/support/articles-and-how-tos/rfp-library>
- + General tips: <http://www.techsoup.org/support/articles-and-how-tos/overview-of-the-rfp-process>
- + Major consideration: scope creep!
- + Project One Pager: <http://www.slideshare.net/tsierra/the-projectonepager>



<https://flic.kr/p/qN5Lvc>



Now it's your turn!

Practice, practice, practice

Exercise 5

- + You have a problem.
- + You have some money.
- + You need to hire someone.
- + What do you do?

References

- + English4IT. “Top 10 Most Popular Programming Languages.” <http://www.english4it.com/unit/9/reading>.
- + Peters, Chris (2011). “An Overview of the RFP Process for Nonprofits, Charities, and Libraries.” 2011 May 10. <http://www.techsoup.org/support/articles-and-how-tos/rfp-library>
- + Serra, Tito (2011). “The Project One-Pager.” SlideShare presentation. <http://www.slideshare.net/tsierra/the-projectonepager>
- + Trautman, Erik (2015). “Why Learning to Code Is So Damn Hard.” Viking Code School Blog, 2015 Feb 24. <http://www.vikingcodeschool.com/posts/why-learning-to-code-is-so-damn-hard>
- + Yelton, Andromeda (2015). “Coding for Librarians: Learning by Example.” Library Technology Reports, 51:3. <http://dx.doi.org/10.5860/ltr.51n3>