got code?
carrie preston
olc technical services retreat 2014

why should librarians learn basic programming?
optimizing workflows
(andromeda yelton, “why should librarians learn python”)

original record creation in OCLC Connexion:
(OCLC Macro Language)

ebook cataloging in OCLC Connexion
(OCLC Macro Language)

oclc# 832314791
why should librarians learn basic programming?

optimizing workflows

like

editing records prior to loading in the ILS

processing reports from the ILS

batch editing records already in the ILS

batch loading records into the ILS

(Shana Mcdanold, “Why am I learning to code”)

Innovative Millennium ERM/WebOPAC:

make resource subject headings clickable

http://alice.library.ohiou.edu/record=b3242589~S7

http://alice.library.ohiou.edu/record=e1000191~S7

$(document).ready(function() {
  // hyperlink resource subjects
  $('resourceSubjects td').each(function(index) {
    if ($(this).hasClass('resourceInfoData')) {
      $(this).text().trim() + '</a>');
    } });
});
Innovative Millennium Course Reserves/WebOPAC:
add a link to each course reading list that links to appropriate LibGuides, depending on subject matter
http://alice.library.ohiou.edu/search~S7?/rcoms+6350
http://alice.library.ohiou.edu/search~S7?/rbios+1030

Innovative Millennium WebOPAC:
reduce repetitive maintenance to search forms and web site header
http://alice.library.ohiou.edu/

why should librarians learn basic programming?

- optimizing workflows
- improving usability
- communicating with IT and vendors
  “insight, dreaming and creation”

(andromeda yelton, “why should librarians learn python”)
starting points
for catalogers who use OCLC Connexion Client:
OCLC Macro Language for the Complete Beginner
http://www.hahnlibrary.net/libraries/oml/lessons/

starting points
for automating software running on a Windows PC
(like Innovative Millennium):
AutoIt

starting points
for website customization:
HTML / CSS
JavaScript / jQuery

starting points
for general programming concepts, if you don't yet know your specific goals:
Python
(PyMARC)
starting points

for general programming concepts, if you don’t yet know your specific goals:

Python
(or whatever other language is used in the course, book, etc. that works for you)

books
print e
(learn python the hard way: http://learnpythonthehardway.org/book)

starting points

video lectures

http://lynda.com/
http://teamtreehouse.com/

starting points

online course - instructor-guided

http://libraryjuiceacademy.com/
http://oreillyschool.com/
MOOCs (Coursera, Udacity, EdX...)
starting points

interactive, online - no instructor

http://codecademy.com/
http://code.org/
http://www.codecademy.com/courses/marc-viewer

you will have to learn by doing
you will not *instantly* create usable products
you will have to use external resources (google)
punctuation and spacing will matter
you will hit frustration points
there is no free lunch

*google* is your friend

http://w3schools.com/
http://developer.mozilla.org/
http://stackoverflow.com/

you may hear why your starting point of choice
(or the learn-to-code movement in general)
is wrong

keep criticism in perspective
“A common complaint...is that you’ll breeze happily through the beginner material, and then hit a steep curve....The only way to get past it is to persevere. This means you keep trying new things, learning more information, and figuring out, piece by piece, how to build your project.”

(cecily carver, “things I wish someone had told me when I was learning how to code”)

“I’ve found that a big difference between new coders and experienced coders is faith: faith that things are going wrong for a logical and discoverable reason....The path from ‘not working’ to ‘working’ might not be obvious, but with patience you can usually find it.”

(cecily carver, “things I wish someone had told me when I was learning how to code”)

got code?
carrie preston
olc technical services retreat 2014