Lesson 1:

1. Introduction

Dave Evans = Instructor

Goal = Learn about computer science, and build a web search engine.

To build a search engine, we need to find data by crawling web pages, build an index to respond quickly to search queries, and rank pages so the most relevant ones appear first.

Units 1-3 = web crawler

Units 4-5 = respond to queries

Unit 6 = rank results

1. Advice from Sergey Brin

Most important thing is to have a good, fun corpus. Internet, books, email, etc.

1. Unit 1: Getting Started – Extracting the first link.

Start from a seed, find a bunch of links from that original seed.

1. Quiz: First Quiz
   1. Check understanding
   2. Don’t count towards the grade

What is the goal of Unit 1?

Great job! We're going to get started programming, learn some computer science concepts and collect a link from a web page!

1. What is Programming?

Computer needs a program to do something we want.

Python = interpreter

1. Quiz: What Is a Program?
   1. A web browser
   2. The Python interpreter
   3. Calendar app on a phone
   4. The Python code in the class
2. Quiz: First Programming Quiz

print 7 \* 7 \* 24 \* 60

1. Congratulations!
2. Quiz: Language Ambiguity

Natural languages like English can be interpreted in many different ways.

It depends on how you interpret the question. Biweekly can happen either twice a week or once every other week.

Verbosity, natural languages are lengthy for code.

1. Grammar

Noun -> Cookies

John Backus was the lead designer of Fortran programming language in the 1950s at IBM.

1. Backus Naur Form

Purpose = be able to precisely describe exactly the language that is simple and concise.

<Non-terminal> -> replacement

Derivation = starting from non-terminal, follow rules to find just terminals.

1. Quiz: Backus Naur Quiz

Great job! You got it right! Python Eat Cookies and Python Eat Python both follow the Subject Verb Object grammar!

1. Quiz: Python Expressions

Python is stricter than English.

3

(1\*(2\*(3\*4)))

(((7)))

1. Quiz: Speed of Light

print(299792458 \* 100 \* 1.0/1000000000)

1. Processors

They need to be small enough for light to travel through them to make a cycle.

1. Grace Hopper

Wrote COBOL. Compiler does all the work at once, and then runs the program. Opposed to an interpreter like Python, does the work at the same time as running it.

1. Variables

Assignment Statement:

Name = Expression

speed\_of\_light = 299792485 # meters per second

billionth = 1.0 / 1000000000

nanostick = speed\_of\_light \* billionth \* 100

1. Quiz: Variable Quiz

print(speed\_of\_light / cycles\_per\_second) # 0.1110342437037037 meters per processor cycle

1. Variables Can Vary

Can reassign the same variable to a new value.

= means assignment, think of it like an arrow

1. Quiz: Varying Variables Quiz 1

hours = 20

1. Quiz: Varying Variables Quiz 2

Great job! You recognized that we never assign a value to minutes, so there's no way we can use minutes to assign a value to seconds. Thus, we see an error.

1. Quiz: Spirit Age

print(age \* days\_in\_a\_year) # 10220 days roughly

1. Strings

print ‘Hello’

print “Hello”

hello = Howdy

print hello # prints “Howdy”

1. Quiz: Valid Strings

“Ada”

‘ ”Ada’

1. Ada

Augusta Ada King, Countess of Lovelace

Arguably the first computer programmer.

1. Quiz: Hello!!!

name = "Brian Mascitello"

<string> + <string> performs concatenation of strings.

1. Strings and Numbers

print ‘!’ \* 12 = !!!!!!!!!!!!

1. Indexing Strings

‘udacity’[0] selects the zeroth character from the string, ‘u’

‘udacity’[1+1] -> ‘a’

1. Quiz: Same Value

Options 1,2,5

1. Selection Sub Sequences

word = ‘assume’

print word[3:4] # prints ‘u’

print word[2:] # prints ‘sume’

1. Quiz: Capital Udacity

print(s[1:].capitalize())

though ‘U’ + s[2:] works, too.

1. Quiz: Understanding Selection

s=’<any string>’

all but s[:-1]

1. Finding Strings in Strings

<string>.find(<string>) # returns position of target string, -1 if it is not found

1. Quiz: Testing

Great job! You've found the two substrings that evaluate to -1!

1. Quiz: Testing 2

Not s.find(‘s’)

1. Finding with Numbers

string.find(‘something’, position to start search)

1. Quiz: Finding with Numbers Quiz

None

1. String Theory

Winner at string theory!

1. Quiz: Extracting Links

Webpage = long string

Looking for: <a href=”<url>”>

Page = content of web page as string

start\_link = page.find('<a href=')

1. Final Quiz

page =('<div id="top\_bin"><div id="top\_content" class="width960">'

'<div class="udacity float-left"><a href="http://udacity.com">')

start\_link = page.find('<a href=')

end\_link = page.find('>', start\_link)

start\_position = start\_link + len('<a href=') + 1

end\_position = end\_link - 1

url = page[start\_position:end\_position]

print(url)

1. Great Job!