

# Python for Beginners

Part 3

# Any Questions from Part 1 & 2?

- ▶ Part 1 - input & output, # comments
- ▶ Part 2
  - ▶ Escape Sequences `\n \t \a \* \\`
  - ▶ Triple Quoted Strings `“””`
  - ▶ Variables
  - ▶ Math `+ - * /`
  - ▶ String Concatenation using `+` and `,`
  - ▶ Formatting Numbers
    - ▶ `format(number, “.2f”)`

# What We Will Learn in This Lesson (Part 3)

- ▶ import statement
- ▶ random numbers
- ▶ dot notation
- ▶ Branching
- ▶ Writing Simple Game Programs
- ▶ Loops

# Unpredictability Makes Games Fun

Unpredictable ...

Means you don't know what will happen



If you want to write a game program  
how do you make it unpredictable?

# Random Numbers

Series of Numbers that when you look at them  
it is

impossible to predict future values  
based on past or present ones.

NOT Random -> 2, 4, 6, 8, 10, 12, 14

Random -> 19, 879, 34, 44, 234

# Random Numbers on a Computer

- ▶ Computers use formulas to create Random Numbers
- ▶ Programmers create functions that use these formulas
- ▶ A function that creates a Random Number is called a ...

Random Number Generator

Is it really a Random Number?

If Computers use formulas to create  
Random Numbers then...

Can it really be a Random Number?

Is it really a Random Number?

No.

It is actually a ... Pseudo Random Number

Real Random Numbers are hard to  
generate.



# How do we get Random Numbers in Python?

At the top of the program file!

```
import random
```

# Import Statement

- ▶ Modules are python files that have code that is already written.

Don't reinvent the Wheel!

- ▶ Import statements let us use these modules in our programs.



# Random Functions

`random.randint(first, last)`

Generates an integer between first & last values

Example:

`die1 = random.randint(1, 6)`

Generates a number between 1 & 6

1, 2, 3, 4, 5, or 6

What type is die1?

# Dot Notation

random.randint()



This Dot means...

The function `randint()` BELONGS to the  
module `random`

# Random Functions

`random.randrange(value)`

Generates an integer between zero & value

Example:

`die2 = random.randrange(5)`

Generates a number between 0 & 5

0, 1, 2, 3, 4, 5

# Random Functions

How do we get an integer between 1 & 6?

```
die2 = random.randrange(5) + 1
```

Creates an integer from 0-5 then adds 1



# Craps Roller Program

```
C:\Python31\python.exe Window Snip
You rolled a 3 and a 4 for a total of 7

Press the enter key to exit.
```

# Let's Code This Program

1. Open a New File
2. Add Comments at Top
3. Import the Random Module

```
# Your Name  
# Craps Roller  
# Demonstrates random number generation  
  
#import the random module  
import random
```



# Let's Generate 2 Random Numbers between 1 & 6 Using the randint() and randrange() functions

```
# generate random numbers 1 - 6
# these are two different ways to do the same thing

die1 = random.randint(1, 6) #Generates a number between 1 and 6

die2 = random.randrange(6) + 1 # Generates a number between 0 and 5
                                # then adds 1

# add the value of the two dice together
total = die1 + die2
```

## Why do we want integers between 1 & 6?

# Let's Generate 2 Random Numbers between 1 & 6 Using the randint() and randrange() functions

```
# generate random numbers 1 - 6
# these are two different ways to do the same thing

die1 = random.randint(1, 6) #Generates a number between 1 and 6

die2 = random.randrange(6) + 1 # Generates a number between 0 and 5
                                # then adds 1

# add the value of the two dice together (math is fun)
total = die1 + die2
```

## Which random function is better?

# Let's Print out the Results

```
# output result to the player
print("You rolled a", die1, "and a", die2, "for a total of", total)

# pause for player input before exiting console window
input("\n\nPress the enter key to exit.")
```

Save & Run Your Program After you finish typing it all in!



# Craps Roller Program

```
C:\Python31\python.exe
You rolled a 3 and a 4 for a total of 7

Press the enter key to exit.
```

*Your results may differ! 😊*

# Branching



Making a decision to take one path or another.

# If *some condition*



Conditions are True or False

# Conditions are Created by Comparing Values

if password == "secret"

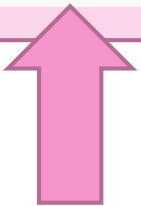


Comparison Operator

When is the value True?  
When is the value False?

# Comparison Operators

<code>x == y</code>	<code># Produce True if ... x is equal to y</code>
<code>x != y</code>	<code># ... x is not equal to y</code>
<code>x &gt; y</code>	<code># ... x is greater than y</code>
<code>x &lt; y</code>	<code># ... x is less than y</code>
<code>x &gt;= y</code>	<code># ... x is greater than or equal to y</code>
<code>x &lt;= y</code>	<code># ... x is less than or equal to y</code>





# Let's Try An Example

1. Type this in a File ... the comments are optional if you need to save time.

```
# print an accurate splash screen advertisement
print("Welcome to Loose Security Inc.")
print("-- where we really don't know anything about security\n")

# prompt the user for his/her password
password = input("Enter your password: ")

# if the password is equal to the string secret
# then it will print the words "Access Granted"
if password == "secret":
    print("Access Granted")

# pause and wait for user input
input("\n\nPress the enter key to exit.")
```

Indented! →

← Colon :

2. Let's Run this Program two times to See What Happens!

On the first try, Enter the password “secret” (no quotes)

On the second try, Enter the password “truth” (no quotes)

# Block Indentation

Do you need multiple instructions to handle your choice?


Try changing your program to print two lines if the password is “secret”

```
# if the password is equal to the string secret
# then it will print the words "Access Granted"
if password == "secret":
    print("Access Granted")
    print("You are very special!")
```

- Indent the SAME Amount!
- Grouped into a Block
- Tabs or Spaces? Just be consistent

# Let's Use An else clause

1. Change your program to use an else clause in the if statement.



```
# if the password is equal to the string secret
# then it will print the words "Access Granted"
if password == "secret":
    print("Access Granted")
# else if the password is not secret then it
# will print "Access Denied"
else:
    print("Access Denied")|
```

2. Let's Run this Program two times to See What Happens!  
On the first try, Enter the password "secret" (no quotes)  
On the second try, Enter the password "truth" (no quotes)

# Got Lots of Choices?

Then use the  
elif  
Clause

**elif means 'else if'**

# Let's Type In Another Example

1. Open a New File and start typing the following... (comments are optional)

```
#import random module
import random

# welcome screen
print("I sense your energy.  Your true emotions are coming across my screen.")
print("You are...")

#generate a random number between 1 and 3
mood = random.randint(1, 3)
```

# Let's Add An If Statement

2. Add the following to your file ... watch out for indenting, colons, and triple quotes!

```
#if mood is equal to 1, print a happy face
if mood == 1:
    # happy
    print( \
        """
        |-----|
        |  o   o  |
        |   <   |
        |  ._.  |
        |-----|
        """)
```

## Let's Add an elif clause

### 3. Add the following to your file ... watch out for indenting, colons, and triple quotes!

```
# else if mood is equal to 2, print a neutral face
elif mood == 2:
    # neutral
    print( \
        """
        -----
        |         |
        |  o   o  |
        |   <   |
        |  ---  |
        |         |
        -----
        """)
```

# Let's Add Another elif Clause

4. Add the following to your file ... watch out for indenting, colons, and triple quotes!

```
# else if mood is equal to 3 print a sad face
elif mood == 3:
    # sad
    print( \
        """
        |-----|
        |  o   o  |
        |   <   |
        |  .'.  |
        |  '.  |
        |-----|
        """)
```



# Let's Finish Up This Example

5. Add the following to your file ...

```
# anything but 1, 2 or 3 and an error is reported, this should never happen
else:
    print("Illegal mood value!  (You must be in a really bad mood).")

# print end message
print("...today.")

# pause and wait for user input
input("\n\nPress the enter key to exit.")
```

6. Now Let's Run this Program to See What Happens!

# Let's See What You Learned!





# The Magic 8 Ball

1. Ask a question out loud.
2. Shake the Magic 8 Ball.
3. Turn it over and look at the message.

## 20 Possible Messages

[https://en.wikipedia.org/wiki/Magic\\_8-Ball](https://en.wikipedia.org/wiki/Magic_8-Ball)

- It is certain
- It is decidedly so
- Without a doubt
- Yes, definitely
- You may rely on it
- As I see it, yes
- Most likely
- Outlook good
- Yes
- Signs point to yes
- Reply hazy try again
- Ask again later
- Better not tell you now
- Cannot predict now
- Concentrate and ask again
- Don't count on it
- My reply is no
- My sources say no
- Outlook not so good
- Very doubtful

Ten of the possible answers are Positive, five are Negative, and five are Neutral.



# The Magic 8 Ball Program

1. Open a New File.
2. Add a comment with your name so everyone knows who wrote this program!
3. Use the `input()` function to ask the user to type their question for the magic 8 ball. We don't need to save it but its traditional to ask.
4. Use one of the `random()` functions to generate a random number between 1 and the number of possible messages. Try to use at least 3 so you can use an `elif` clause.
5. Use `if/elif/else` statements to display at least 3 different messages.
6. (Optional) Use ASCII art to spice it up!
7. Use the `input()` function to pause the output of the program and prompt the user to press the enter key.
8. Save & Run Your Program. Fix any bugs! That's how real programmers like you do it!

# Working Break Time



Write your own Magic Eight Ball Program!

# WARNING!



Code Solution  
Follows  
Don't Peek  
If You Don't Need to!



# The Magic 8 Ball Program

```
finalmagic8ball.py - C:\Users\Daun\Desktop\finalmagic8ball.py (3.4.2)
File Edit Format Run Options Windows Help
# Final Magic 8 Ball Program
# includes all 20 messages

# import the random module
import random

# clear the screen by printing multiple lines
print("\n"*15)

# print out a welcome message to the user
print("===== Welcome to the Magic 8 Ball Program =====")

# ask the user for their question
input("Please Enter Your Question for the Magic 8 Ball.\n")
Ln: 25 Col: 40
```



# The Magic 8 Ball Program

```
finalmagic8ball.py - C:\Users\Daun\Desktop\finalmagic8ball.py (3.4.2)
File Edit Format Run Options Windows Help
# this variable stores a triple-quoted string with an image of
# the magic 8 ball to display before the message
titleImage = """
    ZZOOO88DDNNMM
      7$7$ZZZO88DNNNNMMM+
    Z7II777$ZZO88DDNNMMMMMM
      ,77?:+77$ZOO8DDNNMMMMMM
    $$$7I+=I77$ZOO8DDNNMMMMMM
    OZ$77777$ZZZO88DDNNMMMMMM
    .OOZZ$Z$Z$Z$ZMMMONMMMMMMMMMM
    N8OOZZZZZZZMD      +MMMMMMMMMM$
    D888OOOOOOOM      ZMMMMMMMMMM
=DD8888O8OOM      I8DD888      $MMMMMMMMMM
NNNDD8888M      N      D.      $MMMMMMMMMM
MNNNNNNDDDM      N      .D,      OMMMMMMMMMM
MMMMNNNNNM      8MDDDN.      ZMMMMMMMMMM
MMMMMMMMMM      M      M,      +MMMMMMMMMM
MMMMMMMMMM      M      M,      ,MMMMMMMMMM.
MMMMMMMMMMZ      .~~~~~.      .MMMMMMMMMM.
ZMMMMMMMMMO.      ~MMMMMMMMMM.
.MMMMMMMMMMMMD7~      ,=8MMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
.MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM=
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMM.
"""
Ln: 25 Col: 40
```





# The Magic 8 Ball Program

```
finalmagic8ball.py - C:\Users\Daun\Desktop\finalmagic8ball.py (3.4.2)
File Edit Format Run Options Windows Help

# generate a random number between 1 and 20
# this is the message that will be printed
answer = random.randint(1, 20)

# check the answer value and then print a message
# assigned to that value. I randomly assigned the messages.
# before I print the answer I print the image of the 8 ball
if answer == 1:
    print(titleImage)
    print("It is certain")
elif answer == 2:
    print(titleImage)
    print("It is decidedly so")
elif answer == 3:
    print(titleImage)
    print("Without a doubt")
elif answer == 4:
    print(titleImage)
    print("Yes, definitely")
elif answer == 5:
    print(titleImage)
    print("You may rely on it")
elif answer == 6:
    print(titleImage)
    print("As I see it, yet")
elif answer == 7:
    print(titleImage)
    print("Most likely")
elif answer == 8:
```

Ln: 25 Col: 40



# The Magic 8 Ball Program

```
finalmagic8ball.py - C:\Users\Daun\Desktop\finalmagic8ball.py (3.4.2)
File Edit Format Run Options Windows Help

elif answer == 8:
    print(titleImage)
    print("Outlook good")
elif answer == 9:
    print(titleImage)
    print("Yes")
elif answer == 10:
    print(titleImage)
    print("Signs point to yes")
elif answer == 11:
    print(titleImage)
    print("Reply hazy try again")
elif answer == 12:
    print(titleImage)
    print("Ask again later")
elif answer == 13:
    print(titleImage)
    print("Better not tell you now")
elif answer == 14:
    print(titleImage)
    print("Cannot predict now")
elif answer == 15:
    print(titleImage)
    print("Concentrate and ask again")
elif answer == 16:
    print(titleImage)
    print("Don't count on it")
elif answer == 17:
    print(titleImage)
    print("My reply is no")
elif answer == 18:
    print(titleImage)
    print("My sources say no")
elif answer == 19:
    print(titleImage)
    print("Outlook not so good")
elif answer == 20:
    print(titleImage)
    print("Very doubtful")
# just in case I have a bug I catch all other values here
else:
    print("Oops! The Magic 8 Ball didn't work!")

# make sure that the user gets to see their message
input("\n\nPress Enter to Exit")

Ln: 25 Col: 40
```

# LOOPS

- ▶ Loops are everywhere!
- ▶ Repeat an action
- ▶ Until a condition is true



Lather. Rinse. Repeat.

# While LOOPS

Looks similar to an if statement.

```
while response != "Because.":  
    response = input("Why?\n")
```

# Sentry Variable

```
while response != "Because.":  
    response = input("Why?\n")
```



response is SENTRY variable

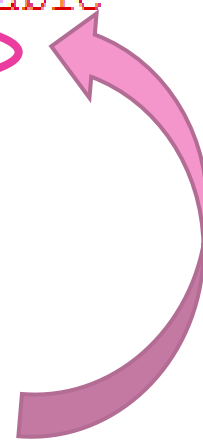
Guards your loop.  
Decides when the loop ends.

# How to determine your sentry variable

```
# 'prime' the loop by clearing out the value  
# in the variable 'response'  
response = ""  
  
# while response is not equal to the word "Because."  
# prompt the user for a new value for response  
# response is the 'sentry variable'  
while response != "Because.":  
    response = input("Why?\n")
```



Know what your stopping  
condition is!



```
# 'prime' the loop by clearing out the value  
# in the variable 'response'  
response = ""
```

```
# while response is not equal to the word "Because."  
# prompt the user for a new value for response  
# response is the 'sentry variable'  
while response != "Because.":  
    response = input("Why?\n")
```

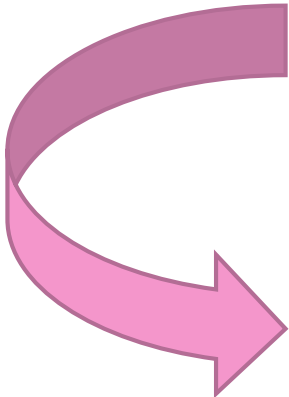


Keeps going while TRUE  
Stops when FALSE




# Setting your sentry variable

Usually Initialize Sentry Variable before using.



```
# 'prime' the loop by clearing out the value  
# in the variable 'response'  
response = ""
```



```
# while response is not equal to the word "Because."  
# prompt the user for a new value for response  
# response is the 'sentry variable'  
while response != "Because."  
    response = input("Why?\n")
```



76 \*ToddlerSimulation.py - C:/Users/Daun/Desktop/ToddlerSimulation.py\*

File Edit Format Run Options Windows Help

```
# Three Year-Old Simulator
# Demonstrates the while loop

# print welcome screen
print("\tWelcome to the 'Three-Year-Old Simulator'\n")
print("This program simulates a conversation with a three-year-old child.")
print("Try to stop the madness.\n")

# 'prime' the loop by clearing out the value
# in the variable 'response'
response = ""

# while response is not equal to the word "Because."
# prompt the user for a new value for response
# response is the 'sentry variable'
while response != "Because.":
    response = input("Why?\n")

# Once the user has typed in "Because."
# this end message will print
print("Oh.  Okay.")

# Keep console open until user presses the enter key
input("\n\nPress the enter key to exit.")
```

Ln: 3 Col: 0

C:\Python31\python.exe

Welcome to the 'Three-Year-Old Simulator'

This program simulates a conversation with a three-year-old child.  
Try to stop the madness.

Why?

I don't know.

Why?

The sky is blue.

Why?

Because

Why?

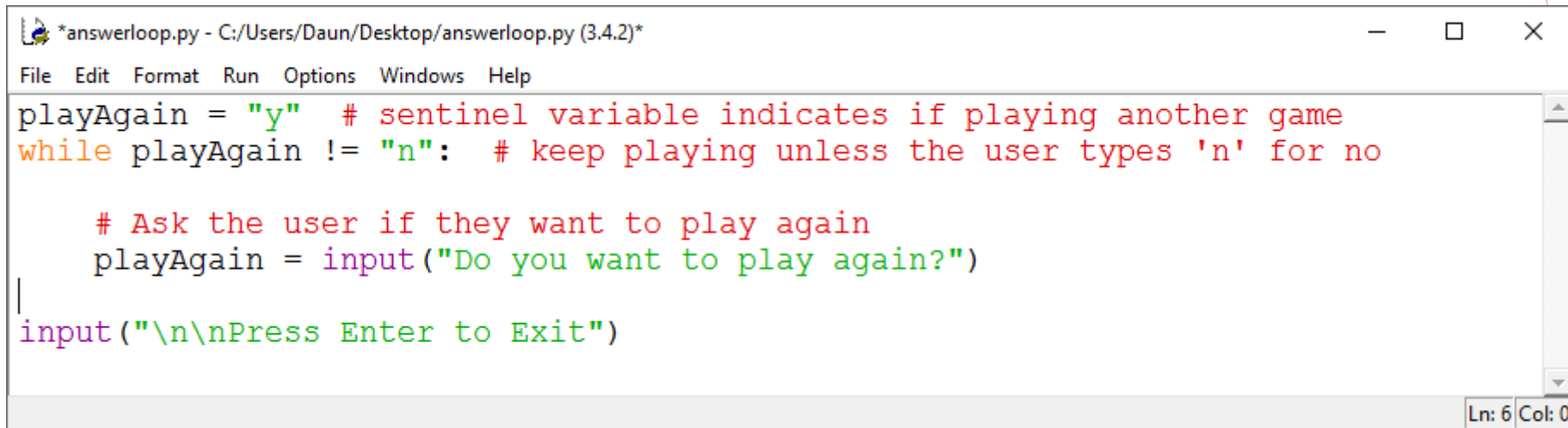
Because.

Oh. Okay.

Press the enter key to exit.

# A Play-Again Loop

Try Typing this Code in A New File and Seeing What it Does!  
Comments are optional to save time!

A screenshot of a Python IDE window titled '\*answerloop.py - C:/Users/Daun/Desktop/answerloop.py (3.4.2)\*'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The code editor contains the following Python code:

```
playAgain = "y" # sentinel variable indicates if playing another game
while playAgain != "n": # keep playing unless the user types 'n' for no


    # Ask the user if they want to play again
    playAgain = input("Do you want to play again?")

input("\n\nPress Enter to Exit")
```

The code is color-coded: strings are in green, comments are in red, and keywords like 'while' and 'input' are in blue. The status bar at the bottom right shows 'Ln: 6 Col: 0'.

# A Play-Again Loop

How Would you Add a Loop to the Magic 8 Ball Program?



```
answerloop.py - C:/Users/Daun/Desktop/answerloop.py (3.4.2)
File Edit Format Run Options Windows Help
# Final Magic 8 Ball Program with A LOOP
# includes all 20 messages

# import the random module
import random

playAgain = "y" # sentinel variable indicates if playing another game
while playAgain != "n": # keep playing unless the user types 'n' for no

    # YOUR ORIGINAL MAGIC 8 BALL GAME GOES HERE
    # NOTICE HOW THIS IS BLOCK INDENTED!

    # Ask the user if they want to play again
    playAgain = input("Do you want to play again?")

# make sure that the user gets to see their message
input("\n\nPress Enter to Exit")

Ln: 11 Col: 40
```

# Working Break Time



ADD A LOOP to your Magic Eight Ball Program!  
Use the Play-Again Loop Example to help you.

# Working Break Time



If You Have Time...  
Can You Add a Loop to one of the other Examples?

# Any Questions from Part 1 & 2?

- ▶ Part 1 - input & output, # comments
- ▶ Part 2
  - ▶ Escape Sequences `\n \t \a \* \\`
  - ▶ Triple Quoted Strings `“””`
  - ▶ Variables
  - ▶ Math `+ - * /`
  - ▶ String Concatenation using `+` and `,`
  - ▶ Formatting Numbers
    - ▶ `format(number, “.2f”)`

# What We Will Learn in This Lesson (Part 3)

- ▶ import statement
- ▶ random numbers
- ▶ dot notation
- ▶ Branching
- ▶ Writing Simple Game Programs
- ▶ Loops



# Recommended Resources

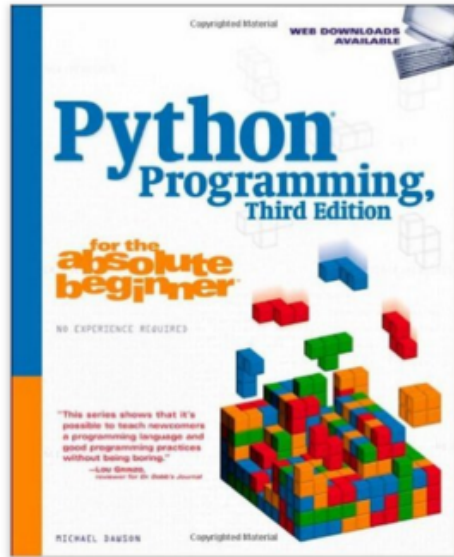
This is the Textbook that I used for this course.

## Python Programming for the Absolute Beginner, 3rd Edition 3rd Edition

by Michael Dawson ▾ (Author)

★★★★☆ ▾ 148 customer reviews


Look inside ↴



ISBN-13: 860-1200556445

ISBN-10: 1435455002

[Why is ISBN important?](#) ▾

eTextbook   
\$2.99

**Paperback**  
\$12.68 - \$19.85

Other Sellers  
from \$12.79

☐ Rent

✓Prime \$12.68

☐ Buy used

✓Prime \$16.79

☒ **Buy new**

✓Prime **\$19.85**

**In Stock.**

List Price: ~~\$34.99~~ Save: \$15.14 (43%)

Ships from and sold by Amazon.com. Gift-wrap available.

55 New from **\$18.64**

✓Prime

Qty: 1 ▾



Add to Cart

or 1-Click Checkout



Buy now with 1-Click®

Order within **1hr 17min** to get it:

**FREE** **1-Click**

# Recommended Resources

<https://www.python.org/about/gettingstarted/>

<https://wiki.python.org/moin/BeginnersGuide>

<https://www.coursera.org/learn/python>

<https://www.udacity.com/course/programming-foundations-with-python--ud036>

The image shows two screenshots of online course pages. The top screenshot is from Coursera for the course 'Programming for Everybody (Getting Started with Python)'. It features a sidebar with links to Overview, Preview, FAQs, and Ratings and Reviews. The main content area includes an 'Enroll Now' button and a description of the course. The bottom screenshot is from Udacity for the course 'Programming Foundations with Python'. It displays the course title, a description, and a 'Start Free Course' button. Below the button, it indicates the course is for beginners, takes approximately 6 weeks, and has 118,726 students. A 'Free' label is also visible at the bottom.

Overview

Preview

FAQs

Ratings and Reviews

Programming for Everybody (Getting Started with Python)

**Enroll Now**  
Starts Apr 11

Home > Computer Science > Software Development

## Programming for Everybody (Getting Started with Python)

**About this course:** This course aims to teach everyone the basics of programming computers using Python. We cover the basics of how one constructs a program from a series of simple instructions in Python. The course has no pre-requisites and avoids all but the simplest mathematics. Anyone with moderate computer experience should be able to master the materials in this course. This course will cover Chapters 1-5 of the textbook "Python for Informatics". This course is equivalent to the first half of the 11-week "Programming for Everybody (Python)" course. Once a student completes this course, they will be ready to take more advanced programming courses.

## Programming Foundations with Python

Learn Object-Oriented Programming

beginner

Approx. 6 weeks  
Assumes 6hrs/wk (work at your own pace)

Join 118,726 students

Start Free Course

START FREE COURSE

Free

# Recommended Resources

## AIMS Community College - CSC 119

### CSC - Computer Science

#### CSC 119 - Introduction to Programming

Focuses on a general introduction to computer programming. Emphasizes the design and implementation of structured and logically correct programs with good documentation. Focuses on basic programming concepts, including numbering systems, control structures, modularization, and data processing. A structured programming language is used to implement the student's program designs. (This class replaces CSC 116.) Three credits.

Department contact: WSTV 247, 339-6244. Click the CRN (5-digit number) to view important additional class section information.

Attendance is required the first day of class to ensure a seat in all computer classes.

Status	Crn	Sect.	Cred	Meeting Time			Location	Cap	Act	Rem	Instructor	Date
OPEN	20355	G11	3.00	M	W	F	10:10am - 11:00am	20	0	20	Bailey, Jennifer	08/22-12/09
OPEN	20356	G12	3.00	M			05:45pm - 08:15pm	20	2	18	McDaniel, Kenneth	08/22-12/09

<http://www.meetup.com/PyNoCo/>

**PyNoCo**

Home Members Sponsors Photos Discussions More • My profile

python

**Fort Collins, CO**  
Founded Mar 14, 2013

About us... Invite friends

Members 262

PyNoCo brings together folks that are interested in Python in Northern Colorado to discuss, share, and show off technologies they are knowledgeable about or interested in learning more about.

Our goal is to foster growth in the use of Python in Northern Colorado, get developers together to share knowledge, and to create a community of developers who have the same passions and interests.

We meet on the third Monday of each month at 5:30pm.

Join members anytime at #pynoco IRC channel on freenode.

# Recommended Resources

Google “Python Tutorials”

# Thank You for Attending!



@gdifortcollins

**Girl Develop It**  
*don't be shy. develop it.* 

Thank You to Our Sponsor  
**Digital Workshop Center**



[digitalworkshopcenter.com](http://digitalworkshopcenter.com)