DURHAM COLLEGE

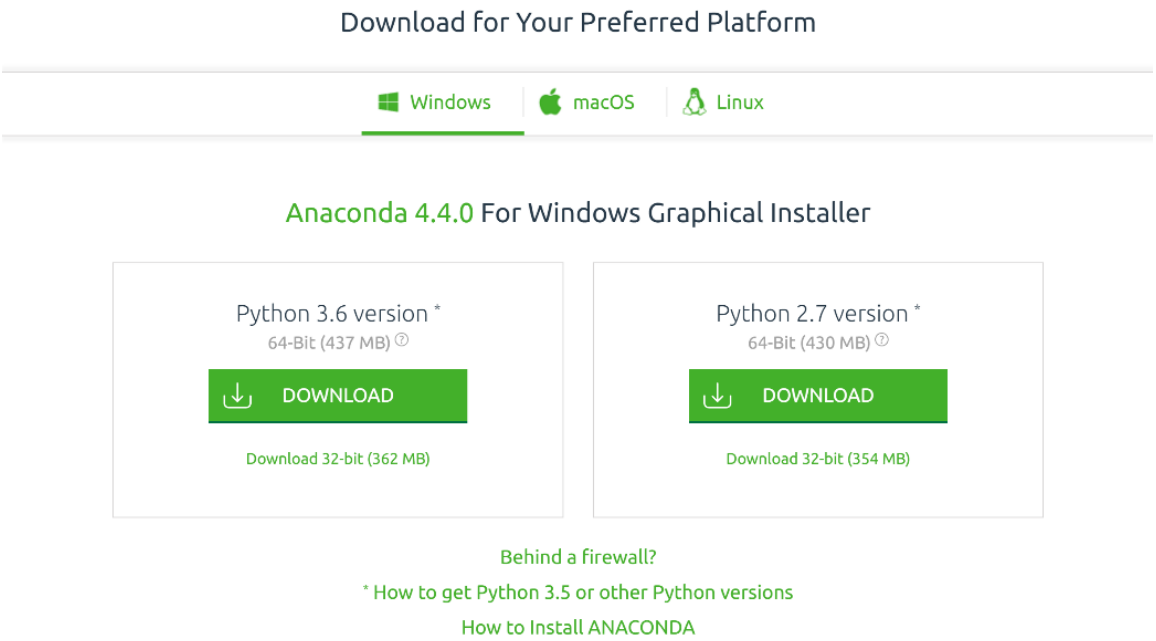
SUCCESS MATTERS

Lesson 1 – Implementation & Python Basics

Background: For this boot camp, we will be using the Anaconda - Python distribution as a foundation for developing and learning the core principles of an A.I. or machine-learning project. The first step is understanding the implementation process of Python on your computer.

Download Anaconda

1. Open a web browser and go to the following link: https://www.continuum.io/downloads
2. Scroll down and select your current Operating System.
3. Select the newest Python version available for download. (i.e. 3.x over 2.x)



Install Anaconda

1. Run the install file once downloaded.
    a. Leave all installation settings as default!
    b. Ensure that Anaconda is selected as your default Python 3.x provider near the end of the installer steps, just before commencing installation.

Exercise Your Python

Read the following information carefully:

- If you need a Python reference at any point, use this link: https://learnxinyminutes.com/docs/python3/
- Remember - object-oriented languages share similar syntax, experiment!
- Python files use the filename extension: '.py'
- Whereas, Python Notebooks use the extension: '.ipny'
    - You will encounter notebooks in Lesson 2
    - *For this lesson, you will be creating a '.py' file.*

This is potentially your first exposure to Python code that someone else has wrote and iterated. In Python, these formally-distributed code packages are called modules.

For this exercise, we need to use a module that can generate random numbers.

- To use a module in your application; At the top of your file, type:
    - 'import random'

This allows your Python program to use a module called 'random' in the rest of your code.

- To generate a random integer and store it as a variable called 'a', type:
    - 'a = random.randint(2, 6)'

Once you run this program, the variable 'a' will have a random integer that the computer generated, between 2 and 6 inclusive. Specifics can be found at the reference link provided above.

Note: There are many ways you can generate random numbers - integers, decimals, and much more. The Python documentation has detailed information as to the random modules capabilities and limitations.

To demonstrate the information provided above; Create a new Python file that does the following:

1. Generate a random number between 1 and 9 inclusive.
2. Prompt the user to guess the number.
3. Tell them whether they guessed ‘too low’, ‘too high’, or ‘exactly right’.
    a. Feel free to be creative.
4. Keep the game going until the user types ‘exit’.
5. Keep track of how many guesses the user has made.
6. When the game ends, print this information out.