

# Day 3: Pre-Class Assignment: Lists and Loops

✔ Put your name here

## Goals for Today’s Pre-Class Assignment

By the end of this assignment, you should be able to:

- Create and manipulate lists in Python
- Write **for** loops and **while** loops in Python
- Combine lists and loops to compute new values.

## Assignment instructions

Read this notebook, watch the videos below and complete the assigned programming problems. Please get started early, and come to office hours if you have any questions!

Recall that to make notebook cells that have Python code in them do something, hold down the ‘shift’ key and then press the ‘enter’ key (you’ll have to do this to get the YouTube videos to run). To edit a cell (to add answers, for example) you double-click on the cell, add your text, and then enter it by holding down ‘shift’ and pressing ‘enter’.

**This assignment is due by 11:59 p.m. the day before class**, and should be uploaded into the appropriate “Pre-class assignments” submission folder on D2L. Submission instructions can be found at the end of the notebook.

## Keyboard shortcuts!

If you want to start using keyboard shortcuts in Jupyter Notebooks to make your life easier and more efficient, **try pressing ESC to enter command mode and then press H**. This will bring up the list of the keyboard shortcuts. Some of the useful shortcuts include **A** and **B**, which insert cells into the notebook “above” and “below” the current cell, respectively. **X** is also useful for cutting out cells that you don’t need.

## Part 1: Lists in Python

One very useful data type/container in Python is the “list”. A list is considered a “mutable” Python data type which means that once you create a list, you are free to change the contents of that list. Lists are also handy because they can store pretty much any other type of python variable within them – including other lists! Creating lists of lists can actually be extremely handy at times. The following table should serve as a simple reference guide for lists. Note that you might have to scroll side to the side to see the whole table.

Container Type	Mutable or Immutable	Initialization Without Values	Initializtion With Values	Adding Values to Container	Removing Values from Container
List	Mutable	<ul style="list-style-type: none"><li>• <code>a=list()</code></li><li>• <code>a=[]</code></li></ul>	<code>a=['1', '2', '3']</code>	<ul style="list-style-type: none"><li>• <code>list.append(item)</code> #Adds item to the end of the list</li><li>• <code>list.insert(index, item)</code> #Adds item to the specified index in the list</li></ul>	<code>list.remove(item)</code> #removes the first instance of 'item' from the list. If there is not such an element, this will cause an erro

**Now**, watch the following video about lists. If the YouTube video doesn’t work, try this link: [https://mediaspace.msu.edu/media/The+list+data+type+in+Python/0\\_55bjfjvn](https://mediaspace.msu.edu/media/The+list+data+type+in+Python/0_55bjfjvn)

```
# Imports the functionality that we need to display YouTube videos in a Jupyter notebook.
# You need to run this cell before you run ANY of the YouTube videos.
from IPython.display import YouTubeVideo

# Watch this video to learn about the list type in Python
YouTubeVideo("TJ_bGrigAMg",width=640,height=360)
```

We can think of a list like multiple containers placed side by side in a line. Each container in our list has an *index* that tells us where that container lies in relation to the other containers in the list. Within each container lies a piece of data that could be any Python variable type. We typically refer to these pieces of data as the *values* of the list.

☰ Contents

✔ Put your name here

- [Goals for Today's Pre-Class Assignment](#)
- [Assignment instructions](#)
- [Keyboard shortcuts!](#)
- [Part 1: Lists in Python](#)
- [Part 2: For loops and While loops](#)
  - [Part 2.1: For loops](#)
  - [Index vs Value](#)
  - [Part 2.2: While loops](#)
- [Part 3: Combining lists and loops.](#)
- [Follow-up Questions](#)
- [Assignment wrap-up](#)
  - [Congratulations, you're done!](#)

Here’s an example list: `[7, 24.5, "class"]`. We can think about this list as three containers next to each other with each container containing a single element from this list. So for this list, we have a container at the index 0 with the value of 7, a container at the index 1 with the value 24.5, and a container with the index 2 with the value “class” (remembering Python indexing starts at 0!).

**Question 1:** In the cell below, create a list that contains, in this order:

1. Your first name as a string
2. Your age as a floating-point number in years. Include one decimal point of precision and a comment about how you calculated the decimal value.
3. Your room, apartment, or house number as an integer

Then:

- Print this list out.
- Replace your first name in the list with your last name, and replace your age with the current year as an integer.
- Append one or more new variables (anything you like), print out the length of the list using the `len()` function, and then print out the entire list again.

```
# Write your program here, using multiple cells if necessary. Add extra cells using
# the 'Cell' menu at the top of this notebook or by using the keyboard shortcut. Don't
forget that you can execute
# your program by holding down 'shift' and pressing 'enter' in each cell!
```

## Part 2: For loops and While loops

### Part 2.1: For loops

Watch the following video, which introduces the concepts of `for` loops and how they work in Python. After you watch the video, answer the questions below.

If the YouTube video below doesn’t work, try this link: [https://mediaspace.msu.edu/media/Loops+in+python/0\\_ls7psyg2?ed=629](https://mediaspace.msu.edu/media/Loops+in+python/0_ls7psyg2?ed=629)

```
# Video on "for" loops in Python
# Make sure to watch it in full-screen mode!
YouTubeVideo("VnTN5sFIPD0",width=640,height=360,end=629)
```

**Question 2:** Write a `for` loop that prints the numbers 0 through 9 in the cell below.

```
# put your code here!
```


**Question 3:** Write a `for` loop that iterates through the items in the “cmse\_grab\_bag” list provided in the cell below and prints each item.

```
cmse_grab_bag = ["CMSE", 201, "Section 001", "Room 3201",
                "pre-class", 3, "in-class", "TA", "Section 005",
                "LA", "flipped classroom"]

# put your code here!
```

**Task:** Looking at Questions 2 and 3, answer the following questions:

- What type of information did you loop over? Explain the code that you wrote in plain English.
- Are there differences between the structure of the loops that you wrote for Questions 2 and 3? If not, how else could you have completed Question 3 such that the structure would be different?

 Put your answer here

### Index vs Value

Remember the container idea we mentioned above?

In `for` loops, we can access the values in a list *by the value itself or by the index of the container*. For example, if you answered Question 3 with the structure of `for value in list:`, you accessed the values each container directly. We could also access the values in the container by using the index of where the container lies in list - in some sense, we are saying “find the container at this index and return the value in that container.”

Looking at the list in Question 3, let’s say we want to know which indices contain a section name. So, now we need to know the index **and** the value of the list at that index. In order to use the index to look through our list, we need to know how many total containers there are in the list. And since indices are numbers, we can easily use the range function from above. Since we want to loop through all the containers in the list, we start at 0 and end at the total number of containers in the list. Then, we print the index and the value of the list at that index.

Here’s what that looks like:

```
for i in range(0, len(cmse_grab_bag)): #this can also be written for i in
range(len(cmse_grab_bag)):
    print(i, cmse_grab_bag[i])
```

We can see that the two section names are located at index 2 and index 8. While this is a simple example, it can easily be expanded to other situations.

Now you might be asking, when do I loop by index and when do I loop by the value? Here’s some tips:

- If you only need to access the values in a single list, without any of the index information, the `for value in list:` structure is most likely the way to go.
- As your code gets more complicated, we often use the `for i in range(len(list)):` structure. Some examples are:
  - When you need the index information, like above
  - When you want to access values at the same index in multiple lists

The gif below shows the difference between looping over the value versus the index. Notice that the lists and the output for each are the same.


### Loop by Value

```
subject_list = ['math', 'computer science',  
               'chemistry', 'English', 'history']  
  
for item in subject_list:  
    print(item)  
  
Step 1: 'item' = 'math'
```

### Loop by Index

```
subject_list = ['math', 'computer science',  
               'chemistry', 'English', 'history']  
  
for i in range(len(subject_list)):  
    print(i, subject_list[i])  
  
Step 1: 'i' = '0'  
       'subject_list[i]' = 'math'
```

 **Task:** Before moving on to while loops, explain the idea of an index to a someone who has never coded before.

 Put your answer here


## Part 2.2: While loops

Watch the following video clip, which introduces the concept of `while` loops in Python. After you watch the video, answer the questions below.

**Note:** Ignore the little bit at the end of the video clip that mentions numpy arrays and the `nditer` function.


If the YouTube video below doesn’t work, try this link: [https://mediaspace.msu.edu/media/Loops+in+python/0\\_ls7psyg2?st=823](https://mediaspace.msu.edu/media/Loops+in+python/0_ls7psyg2?st=823)


```
# Video on "while" loops in Python  
# Make sure to watch it in full-screen mode!  
YouTubeVideo("VnTN5sFIPD0",width=640,height=360,start=823)
```

 **Question 4:** Write a `while` loop that starts with a value of “`x = 2`”, on each iteration multiples `x` by 2, prints the value of `x`, and keeps running until `x` is greater than 4096.


**Caution:** When working with while loops it is easy to accidentally create an “infinite” loop that runs forever. If you do this the notebook will get stuck on a cell (appearing as `In [*]:` on the left). The best way to fix this is to select “Interrupt” from the Kernel menu above.

```
# put your code here!
```

 **Task:** Now that you have worked with both for and while loops, what are the main differences between them? How would you explain the difference to a friend who did not get to watch the awesome videos in this lesson?

 Put your answer here

## Part 3: Combining lists and loops.


 **Question 5:** Now we want you to try to use various components of everything you just learned to calculate some values. The cell below includes two lists of values, `x` and `y`. Your task is to initialize two new empty lists, `xsquared` and `ycubed`, iterate through the two lists of values, `x` and `y`, using loops to compute  $x^2$  and  $y^3$ , and store these values in your new lists. Once you’ve computed all the values, print out the results.

**Yes**, this might seem a bit complicated – that’s totally expected. After all, you’re still learning to write code! If you find yourself spending too much time to figure this out, make use of Slack or try to drop by office hours or help room hours. Making sure that you spend time on your own trying to work through pre-class activities like this one are important for making your time in class as productive as possible.

**Hint:** If you take a look at the table in Part 1 that contains information about lists, you see that it mentions adding values to a list using `.append()`, this is what you’ll want to use to fill your `xsquared` and `ycubed` lists. Also, remember that we use `**` to compute powers in Python. So “2 cubed” would be `2**3`.

Before you write any code, answer these questions:

- What information do you need from each list?
- How are you going to access that information?
- Once you have the information, what are you going to do with it?

 Put your answer here

```
x = [2,4,6,8,10,12,14,16,18]
y = [10,8.25,7.5,7,6.5,7,7.5,8.25,10]

# Put your commands for intializing your new lists below this comment

# Put your loop(s) for computing x-squared and y-cubed below these comments.
# You'll be using .append() to add new values to the lists you just initialized.
```

## Follow-up Questions

Copy and paste the following questions into the appropriate box in the assignment survey include below and answer them there. (Note: You'll have to fill out the section number and the assignment number and go to the "NEXT" section of the survey to paste in these questions.)

1. If you had a list variable called `directory`, how you would access the fourth element in this list?
2. Explain when you might use a `for` loop versus a `while` loop.

Print to PDF ►

## Assignment wrap-up

Please fill out the form that appears when you run the code below. **You must completely fill this out in order to receive credit for the assignment!**

```
from IPython.display import HTML
HTML(
    """
    <iframe
        src="https://cmse.msu.edu/cmse201-pc-survey"
        width="800px"
        height="600px"
        frameborder="0"
        marginheight="0"
        marginwidth="0">
        Loading...
    </iframe>
    """
)
```

## Congratulations, you're done!

Submit this assignment by uploading it to the course Desire2Learn web page. Go to the "Pre-class assignments" folder, find the appropriate submission folder link, and upload it there.

See you in class!

© Copyright 2021, Michigan State University Board of Trustees

By MSU CMSE  
© Copyright 2021.