Elevator

July 16, 2023

1 Practice Notebook: Methods and Classes

The code below defines an *Elevator* class. The elevator has a current floor, it also has a top and a bottom floor that are the minimum and maximum floors it can go to. Fill in the blanks to make the elevator go through the floors requested.

This class is pretty empty and doesn't do much. To test whether your *Elevator* class is working correctly, run the code blocks below.

```
In [112]: class Elevator:
              def __init__(self, bottom, top, current):
                  """Initializes the Elevator instance."""
                  self.bottom = bottom
                  self.top = top
                  self.current = current
              def __str__(self):
                  return "Current floor: {}".format(self.current)
              def up(self):
                  """Makes the elevator go up one floor."""
                  if self.current < 10:
                      self.current += 1
              def down(self):
                  """Makes the elevator go down one floor."""
                  if self.current > 0:
                      self.current -= 1
              def go_to(self, floor):
                  """Makes the elevator go to the specific floor."""
                  if floor >= self.bottom and floor <= self.top:
                      self.current = floor
                  elif floor < 0:
                      self.current = 0
                  else:
                      self.current = 10

          elevator = Elevator(-1, 10, 0)
```

```
In [122]: # Go to the top floor. Try to go up, it should stay. Then go down.
          elevator.go_to(10)
          elevator.up()
          elevator.down()
          print(elevator.current) # should be 9
          # Go to the bottom floor. Try to go down, it should stay. Then go up.
          elevator.go_to(1)
          elevator.down()
          elevator.down()
          elevator.up()
          print(elevator.current) # should be 1
```

```
9
1
```

Once you've made the above methods output 1, 0 and 10, you've successfully coded the *Elevator* class and its methods. Great work! For the up and down methods, did you take into account the top and bottom floors? Keep in mind that the elevator shouldn't go above the top floor or below the bottom floor. To check that out, try the code below and verify if it's working as expected. If it's not, then go back and modify the methods so that this code behaves correctly.

```
In [113]: elevator.up()
          elevator.current #should output 1
```

```
Out[113]: 1
```

```
In [107]: elevator.down()
          elevator.current #should output 0
```

```
Out[107]: 0
```

```
In [109]: elevator.go_to(10)
          elevator.current #should output 10
```

```
Out[109]: 10
```

Now add the **str** method to your *Elevator* class definition above so that when printing the elevator using the **print( )** method, we get the current floor together with a message. For example, in the 5th floor it should say "Current floor: 5"

```
In [115]: elevator.go_to(5)
          print(elevator)
```

```
Current floor: 5
```

If you get a **NameError** message, be sure to run the *Elevator* class definition code block first. If you get an **AttributeError** message, be sure to initialize *self.current* in your *Elevator* class.

Remember, Python uses the default method, that prints the position where the object is stored in the computer's memory. If your output is something like: > <**main**.Elevator object at 0x7ff6a9ff3fd0>

Then you will need to add the special **str** method, which returns the string that you want to print. Try again until you get the desired output, "Current floor: 5".

Once you have successfully produced the desired output, you are all done with this practice notebook. Awesome!