# If elif else

## Today we will learn about using if statements to control code and what boolean is.

A boolean tells if something is true or false. True is usually set to 1 and False to 0. lets try a few things.

A double equal sign in Python says it is testing for truth or is a boolean

```
In [1]:   %matplotlib ipympl
          import numpy as np
          import matplotlib.pyplot as plt
```

```
In [2]:   a=1
          b=1
          a==b
```

Out[2]:   True

```
In [3]:   a=1
          b=2
          a==b
```

Out[3]:   False

I just set a and b then asked the computer to tell me if they were equal. That is the double equal signs. Now we need to add "and" and "or". "and" requires both things to be true. "or" only requires 1. Go through all the next code. Guess the answer. Uncomment the code. Run it and see if you are correct.

```
In [5]:   #a==1 and b==2
```

```
In [65]:  #a==1 and b==1
```

```
In [66]:  #a==1 or b==1
```

```
In [7]:   #a==2 or b==1
```

You can also use not to say the opposite... Again try to guess then uncomment the code and see if you get it correct. write down your guess first!

```
In [5]:   #a==1 and not b==1
```

```
In [7]:   #not a==2 or b==1
```

```
In [9]: #not (a==1 or b==1)
```

You have more options

== Checks if equal

!= checks if not equal

<> checks if not equal(this is phased out and will give an error)

> checks if greater than

< checks if less than

>= checks if greater than or equal

<= checks if less than or equal

Again try to guess the answer then go through and check!

```
In [9]: #a>b
```

```
In [17]: #a<b
```

```
In [11]: #a!=b
```

```
In [13]: #a<>b
```

```
In [23]: #a not = b
```

```
In [15]: #(a==b) or (a!=b)
```

```
In [17]: #not ((a==b) or not (a!=b))
```

# Now you should be a boolean expert!!!! remember for long statements do the parentheses first.

Now lets try an if statement

```
In [4]: if a==1:
            print( 'We did our first if statement')
```

```
We did our first if statement
```

## Some common errors!!!

```
In [5]:  If a==1:   #it is lower case if!
```

```
   Cell In[5], line 1
     If a==1:   #it is lower case if!
        ^
SyntaxError: invalid syntax
```

```
In [6]:  if a=1:  # you need the double ==, remember boolean!
```

```
   Cell In[6], line 1
     if a=1:  # you need the double ==, remember boolean!
         ^
SyntaxError: invalid syntax
```

```
In [7]:  if a==1  #you need the :
```

```
   Cell In[7], line 1
     if a==1  #you need the :
             ^
SyntaxError: invalid syntax
```

## now we can make the statement more complicated

```
In [5]:  if a==1 and b==2:
             print ('my second if')
```

```
my second if
```

But what if you want multiple options? You use elif

```
In [35]:  a=2
          if a==1:
              print ('Yeah a=1')
          elif a==2:
              print ('oh no a=2')
```

```
oh no a=2
```

What if you only want to know if it is different?

```
In [37]:  a=3
          if a==1:
              print ('Yeah a=1')
          elif a==2:
              print ('oh no a=2')
          else:
              print ('wow a 3, do we really know it is 3?')
```

```
wow a 3, do we really know it is 3?
```

Remember the else means it could be anything else!

```
In [38]:  a=3000
          if a==1:
              print ('Yeah a=1')
```

```
elif a==2:
    print ('oh no a=2')
else:
    print ('wow a 3, do we really know it is 3?')
```

wow a 3, do we really know it is 3?

# Now lets use input

input takes any input and lets you use it. We are going to assume people enter correctly and do this simply. We will not do any error checking. So first lets get a number with input, then print it.

input sometimes crashes and is a little buggy. But it is good for learning if statements.

if you computer becomes stuck you will see the circle next to Python3 on the top right stay filled.

```
In [8]:  a=input("Please enter an integer from 1 to 3")
         print ('The input was {}'.format(a))
```

The input was 2

Now go back up to that cell and type a letter. See what happens? Nothing it gives you a letter. Lets see what the type is we get from input.

```
In [9]:  a=input("Please enter an integer from 1 to 3")
         print ('The input was {}'.format(a))
         print( type(a))
```

The input was b
<class 'str'>

Try entering a few things above. See what happens to type? try an integer and a float. So we will cast the input to an integer. But that will cause the program to crash if entered wrong. Sometime in the future we may do error catching but for now let's assume people enter perfectly. So cast is an in int and now try an integer,float, and string.

```
In [13]:  a=int(input("Please enter an integer from 1 to 3"))
          print ('The input was {}'.format(a))
          print( type(a))
```

The input was 2
<class 'int'>

Now for fun we can combine our input and if statements!

```
In [14]:  a=int(input("Please enter an integer from 1 to 3"))
          if a==1:
              print ('Yeah a=1')
          elif a==2:
              print ('oh no a=2')
```

```
elif a==3:
    print ('a 3!')
else:
    print ('something else?')
```

oh no a=2

## In coding I don't use input that often. But they can be nice. It can be buggy but I thought it was fun here.

If things stop working you might need to restart your kernal. This happens often with input.

## Let's put for loops and if statements together.

- make a for loop with arange from 0-10.
- print out only when the loop is at 5.

In [15]:
```python
for i in np.arange(11):
    if i==5:
        print ('Wow i={}'.format(i))
```

Wow i=5

Lets make it trickier!

- Using mod print if divisible by 2
- Using mod print if divisible by 5
- Else print not divisible by 2 or 5

In [16]:
```python
for i in np.arange(11):
    if i%2==0:
        print ('Wow i={} and is divisible by 2'.format(i))
    elif i%5==0:
        print ('Wow i={} and is divisible by 5'.format(i) )
    else:
        print ('Wow i={} and is not divisible by 2 or 5'.format(i) )
```

```
Wow i=0 and is divisible by 2
Wow i=1 and is not divisible by 2 or 5
Wow i=2 and is divisible by 2
Wow i=3 and is not divisible by 2 or 5
Wow i=4 and is divisible by 2
Wow i=5 and is divisible by 5
Wow i=6 and is divisible by 2
Wow i=7 and is not divisible by 2 or 5
Wow i=8 and is divisible by 2
Wow i=9 and is not divisible by 2 or 5
Wow i=10 and is divisible by 2
```

## I made two classic mistakes!

- can you make it go from 1-10 so we don't have to worry about 0?
  - can you change the if statements to say if divisible by both 2 and 5?
- the not quite correct answer is below!

In [18]:

```
Wow i=0 and is divisible by both 2 and 5
Wow i=1 and is not divisible by 2 or 5
Wow i=2 and is divisible by 2
Wow i=3 and is not divisible by 2 or 5
Wow i=4 and is divisible by 2
Wow i=5 and is divisible by 5
Wow i=6 and is divisible by 2
Wow i=7 and is not divisible by 2 or 5
Wow i=8 and is divisible by 2
Wow i=9 and is not divisible by 2 or 5
Wow i=10 and is divisible by both 2 and 5
```

# Plot Recent Temperature Data!

This is a long assignment but is interesting. I went to weather underground and got the high temperatures from Laguardia (Staion id KLGA) and Montauk (station id KNYMONTA39) for the first seven days of January. They are listed below. We want to use input and if statements to ask the user what data they want plotted. We could plot

1. Laguardia (KLGA) Temperature versus day in January.
2. Montauk (MONT) Temperature versus day in January.
3. Both KLGA and MONT Temperature versus day in January.
4. KLGA versus MONT to see if they are correlated.

The high temperatures in January at Montauk and Laguardia are

| Day | KLGA | Montauk |
|-----|------|---------|
| 1 | 52 | 53 |
| 2 | 43 | 44 |
| 3 | 39 | 40 |
| 4 | 34 | 37 |
| 5 | 33 | 36 |
| 6 | 31 | 30 |
| 7 | 34 | 32 |

You can go look at my answer and see what I did but you have to do your own. Here is how I would attack it.

1. I would make my array of temperatures. Remember np.array([])?

2. MAKE SURE to add a decimal place to one of your temperatures to make them all floats as good practice for doing math. This will come back and bite you! (Python 3 has fixed this bug but I still add decimals!)

3. I would then use print statements to describe what you are doing and what should be inputted for choices.

4. Then use input to get the number input.

5. I would add the if statements with the plotting in each if statement. You are going to need to also use elif and else.

6. I would make sure to label my graphs well.

7. I would remember how we have done plots in the past. what do we need to import? Our lists start at 0. How do we do the x-axis? Do we need a legend on the one with two lines?

8. For comparing LGA and Montauk you will see that you don't want a line. I would do ax.scatter()

9. Work with your neighbor if you get stuck!

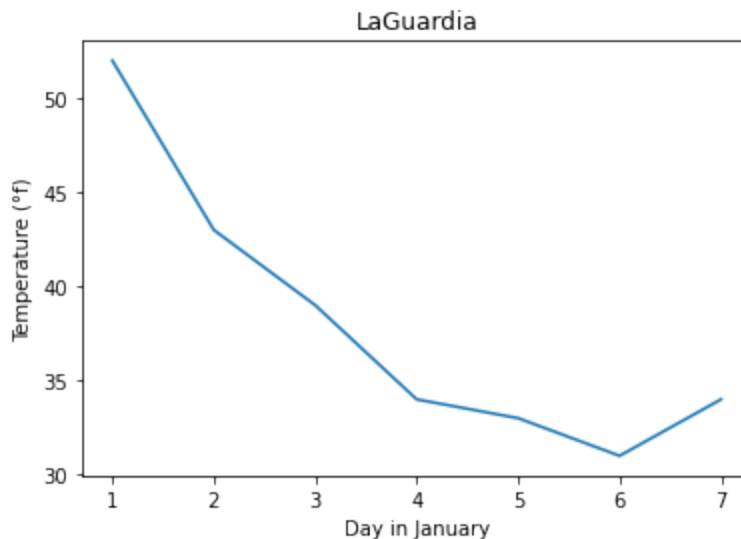10. You can see what my graphs look like below for each of the 4 options

```
In [8]: #Put in your program here
```

```
In [ ]:
```
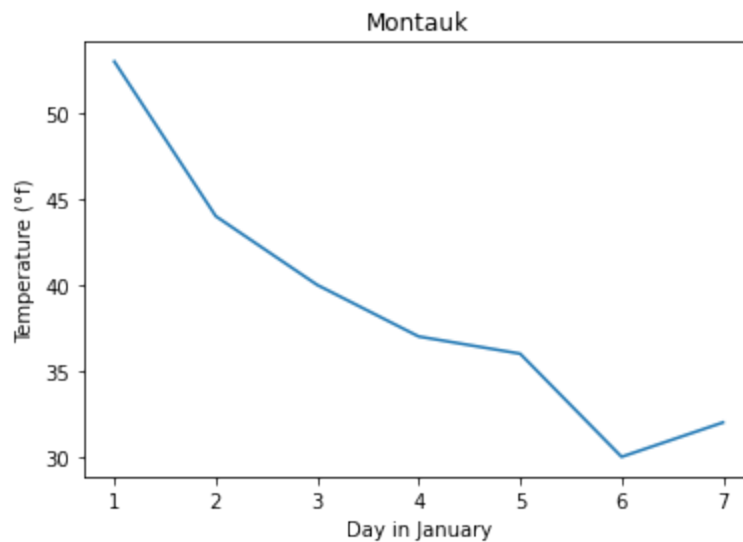
## Option 1

```
In [6]:
```

```
We are going to print Recent Temperature Data!
For LaGuardia type 1
For Montauk type 2
For both on one plot type 3
For a plot of LaGuardia versus LaGuardia type 4
Please enter your number.....1
```
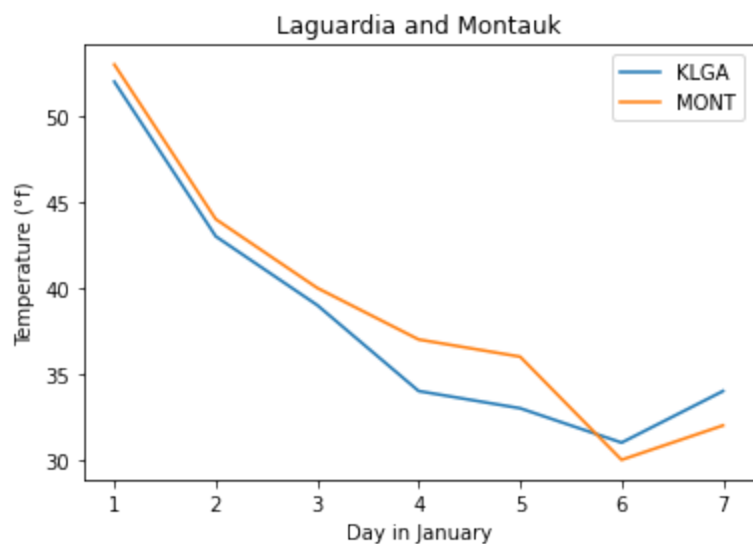
## Option 2

In [7]:

```
We are going to print Recent Temperature Data!
For LaGuardia type 1
For Montauk type 2
For both on one plot type 3
For a plot of LaGuardia versus LaGuardia type 4
Please enter your number.....2
```



## Option 3

In [8]:

```
We are going to print Recent Temperature Data!
For LaGuardia type 1
For Montauk type 2
For both on one plot type 3
For a plot of LaGuardia versus LaGuardia type 4
Please enter your number.....3
```

## Option 4

In [9]:

```
We are going to print Recent Temperature Data!
For LaGuardia type 1
For Montauk type 2
For both on one plot type 3
For a plot of LaGuardia versus LaGuardia type 4
Please enter your number.....4
```
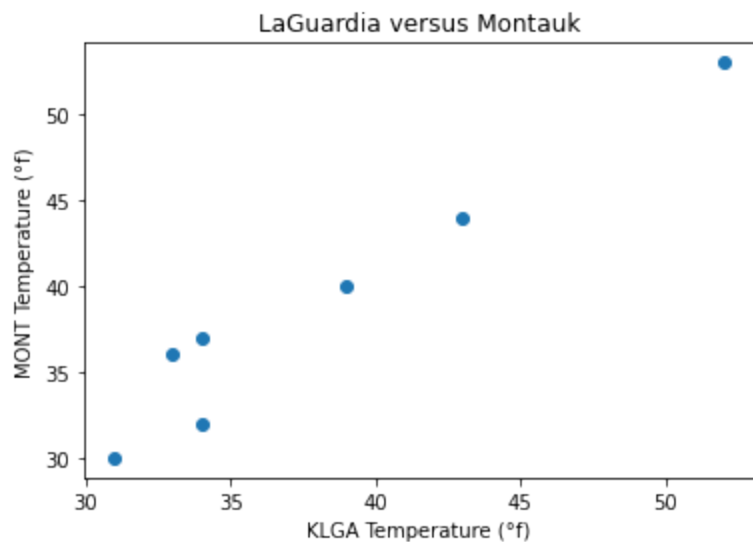


## Answers

In [18]:
```python
for i in np.arange(1,11):
    if i%2==0 and i%10==0:
        print ('Wow i={} and is divisible by both 2 and 5'.format(i))
    elif i%2==0:
        print ('Wow i={} and is divisible by 2'.format(i))
    elif i%5==0:
        print ('Wow i={} and is divisible by 5'.format(i))
```

```
        else:
            print ('Wow i={} and is not divisible by 2 or 5'.format(i))
```

```
Wow i=1 and is not divisible by 2 or 5
Wow i=2 and is divisible by 2
Wow i=3 and is not divisible by 2 or 5
Wow i=4 and is divisible by 2
Wow i=5 and is divisible by 5
Wow i=6 and is divisible by 2
Wow i=7 and is not divisible by 2 or 5
Wow i=8 and is divisible by 2
Wow i=9 and is not divisible by 2 or 5
Wow i=10 and is divisible by both 2 and 5
```

In [22]:
```python
KLGA=np.array([52,43,39,34,33,31,34])
MONT=np.array([53,44,40,37,36,30,32])

print ('We are going to print Recent Temperature Data!')
print ('For LaGuardia type 1')
print ('For Montauk type 2')
print ('For both on one plot type 3')
print ('For a plot of LaGuardia versus LaGuardia type 4')
a=int(input('Please enter your number.....'))

fig,ax=plt.subplots()

if a==1:
    ax.plot(range(1,8),KLGA)
    ax.set_xlabel('Day in January')
    ax.set_ylabel(u'Temperature (\N{DEGREE SIGN}f)')
    ax.set_title('LaGuardia')
elif a==2:
    ax.plot(range(1,8),MONT)
    ax.set_xlabel('Day in January')
    ax.set_ylabel(u'Temperature (\u00b0f)')  #I am adding unicode text with
    ax.set_title('Montauk')
elif a==3:
    ax.plot(range(1,8),KLGA,label='KLGA')
    ax.plot(range(1,8),MONT,label='MONT')
    ax.set_xlabel('Day in January')
    ax.set_ylabel(u'Temperature (\N{DEGREE SIGN}f)')  #I am adding unicode t
    ax.set_title('Laguardia and Montauk')
    ax.legend()
elif a==4:
    ax.scatter(KLGA,MONT)
    ax.set_xlabel(u'KLGA Temperature (\N{DEGREE SIGN}f)')
    ax.set_ylabel(u'MONT Temperature (\N{DEGREE SIGN}f)')
    ax.set_title('LaGuardia versus Montauk')
else:
    print ('you entered a bad number')
```
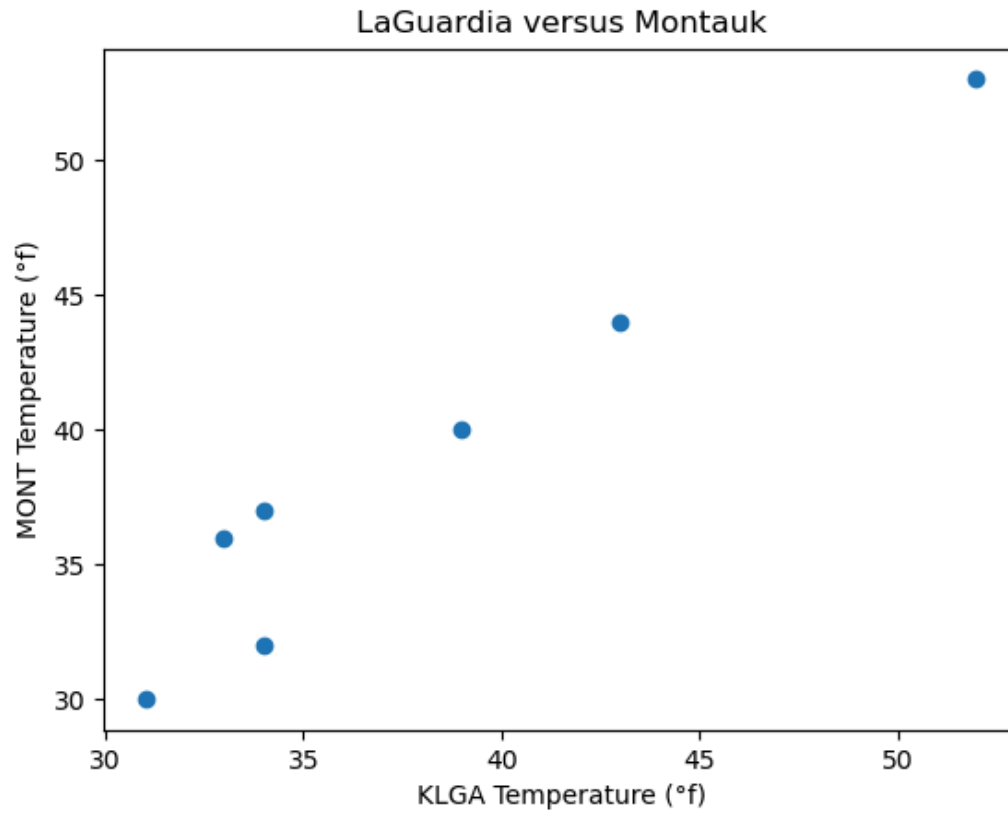
```
We are going to print Recent Temperature Data!
For LaGuardia type 1
For Montauk type 2
For both on one plot type 3
For a plot of LaGuardia versus LaGuardia type 4
```

Figure



```
In [ ]:
```