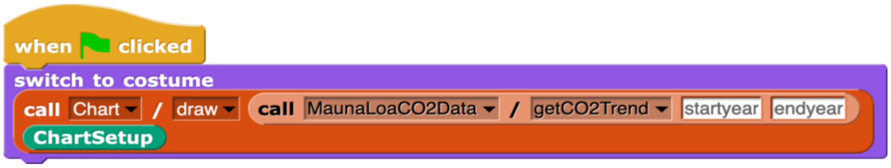
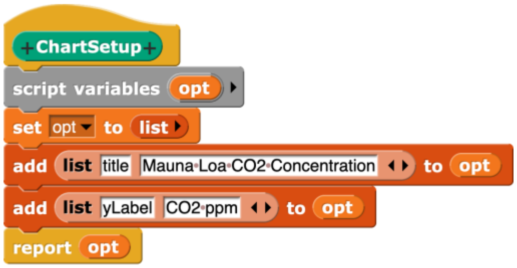
NetsBlox Lesson: Plotting Ice Core Data

In the lesson on plotting, we already utilized climate change related data. We plotted atmospheric CO2 concentrations using data measured at the Mauna Loa Observatory in Hawaii since 1958. Here is a refresher. The service is under the Climate heading in the menu. It has two RPCs, **`getCO2Trend`** and **`getRawCO2`**. The raw data is one measurement per month. Since the CO2 concentration has an annual cycle, the trend is just an annual average value, again once a month. Here is the code that plots the smoothed version of the data:



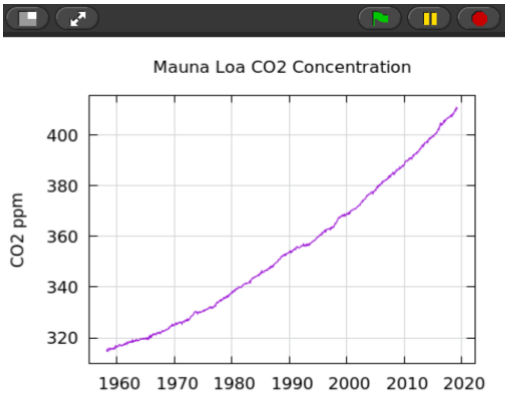
```
when ⚑ clicked
switch to costume (call Chart / draw (call MaunaLoaCO2Data / getCO2Trend startyear endyear) ChartSetup)
```

If the start and stop year are not specified, the RPC returns the data for the entire date range available. The **`ChartSetup`** custom block should look familiar:



```
+ChartSetup+
script variables opt
set opt to list
add (list title Mauna Loa CO2 Concentration) to opt
add (list yLabel CO2 ppm) to opt
report opt
```

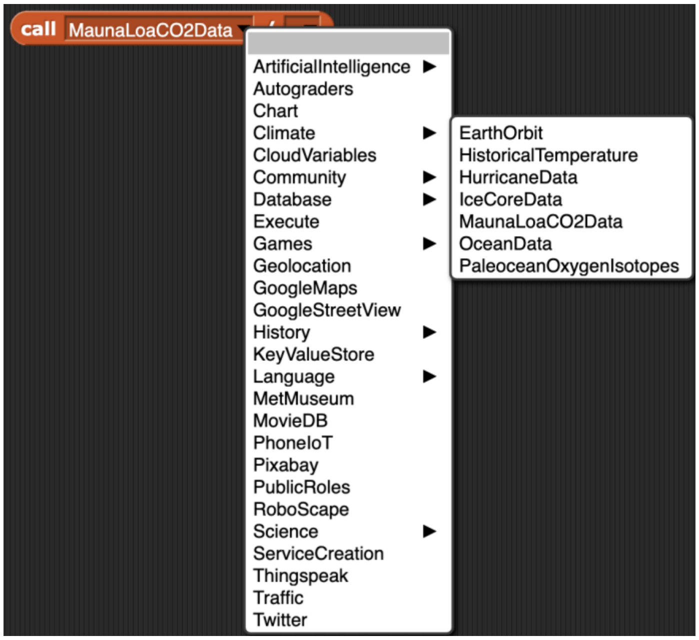
And the result:



The project is available here:
https://editor.netsblox.org/?action=present&Username=ledeczi&ProjectName=Mauna%20Loa%20v2

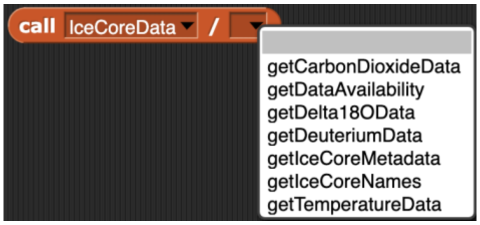
Ice Core Data

NetsBlox has several other data services related to climate change:

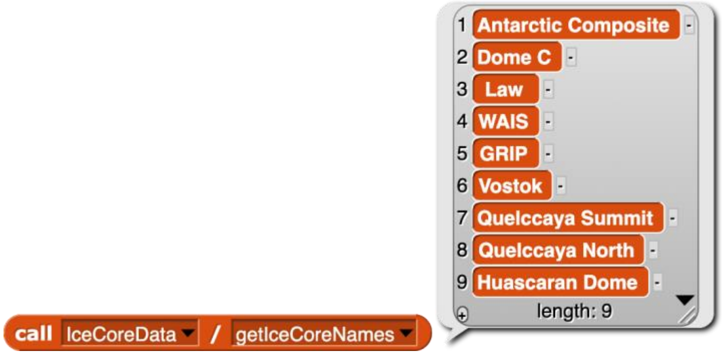


One fascinating data source is ice cores. Researchers have drilled down into ice fields in Antarctica, Greenland and even high mountain glaciers and obtained ice as old as 800,000 years. The ice captured air bubbles that can be analyzed to gain insight into Earth's atmosphere in ancient times. Here is an interesting website that talks about ice cores: https://icecores.org/about-ice-cores

One of the important data that researchers have been able to collect is CO2 concentrations from the past. This is what we will use today. So, let's explore the IceCoreData service. It has the following RPCs:



Let's start by figuring out what ice cores we have data from:

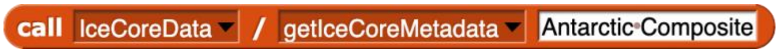


```
call IceCoreData / getIceCoreNames
```

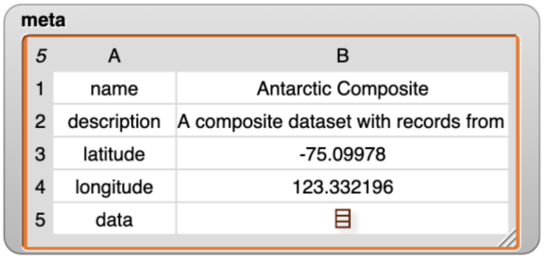
1	Antarctic Composite
2	Dome C
3	Law
4	WAIS
5	GRIP
6	Vostok
7	Quelccaya Summit
8	Quelccaya North
9	Huascaran Dome

length: 9

GRIP is in Greenland, while the last three are from glaciers high up in the Andes Mountains. The rest are from various places in Antarctica. The Antarctic Composite is a data set that was put together from various ice cores like Dome C and Vostok. To investigate what kind of data a particular core has, we can use the `getIcecoreMetadata` RPC:



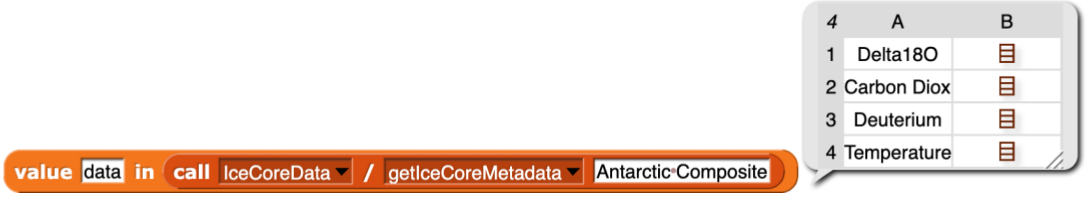
```
call IceCoreData / getIceCoreMetadata Antarctic Composite
```



meta

5	A	B
1	name	Antarctic Composite
2	description	A composite dataset with records from
3	latitude	-75.09978
4	longitude	123.332196
5	data	☰

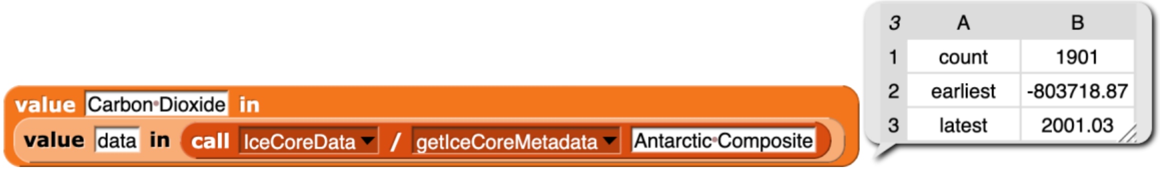
We have the name, a short description, the location and some info about the data itself. As you can see, we have to dig a little deeper to get to that:



```
value data in call IceCoreData / getIceCoreMetadata Antarctic Composite
```

4	A	B
1	Delta18O	☰
2	Carbon Diox	☰
3	Deuterium	☰
4	Temperature	☰

And deeper still:

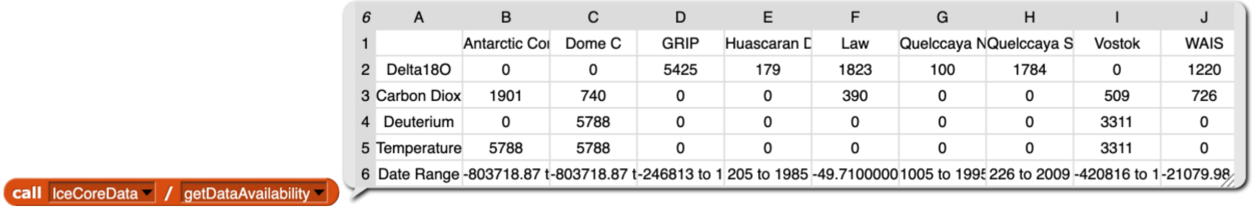


```
value Carbon Dioxide in
  value data in call IceCoreData / getIceCoreMetadata Antarctic Composite
```

3	A	B
1	count	1901
2	earliest	-803718.87
3	latest	2001.03

So, in the Antarctic Composite dataset, we 1901 data points for CO2 concentrations starting over 800,000 years ago and going all the way too early 2001.

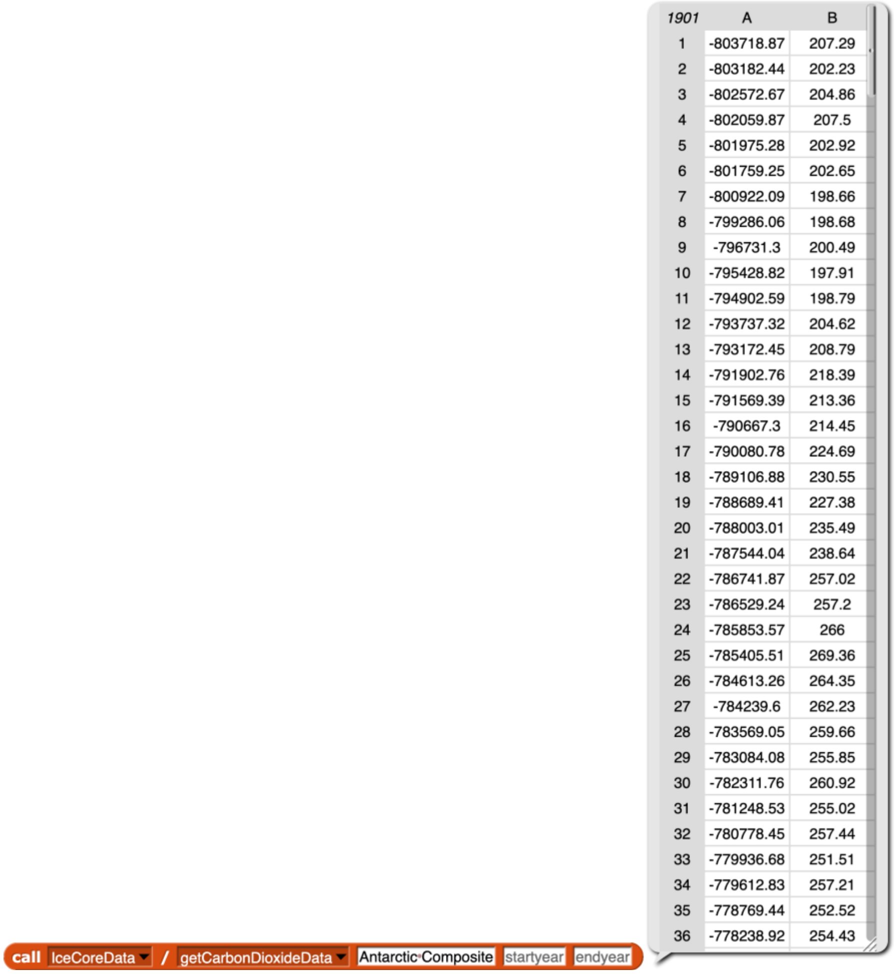
If you want a quick overview of all the data that is available, you can use this RPC:



call IceCoreData / getDataAvailability

6	A	B	C	D	E	F	G	H	I	J
1		Antarctic Coı	Dome C	GRIP	Huascaran D	Law	Quelccaya N	Quelccaya S	Vostok	WAIS
2	Delta18O	0	0	5425	179	1823	100	1784	0	1220
3	Carbon Diox	1901	740	0	0	390	0	0	509	726
4	Deuterium	0	5788	0	0	0	0	0	3311	0
5	Temperature	5788	5788	0	0	0	0	0	3311	0
6	Date Range	-803718.87 t	-803718.87 t	-246813 to 1	205 to 1985	-49.7100000	1005 to 1995	226 to 2009	-420816 to 1	-21079.98

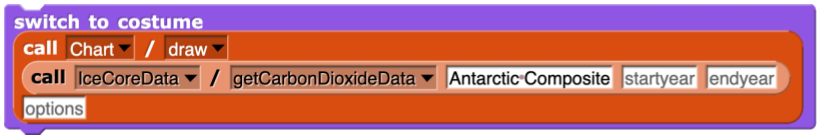
Today, we'll focus on CO2 data from the Antarctic Composite data set. That is easy enough to get:



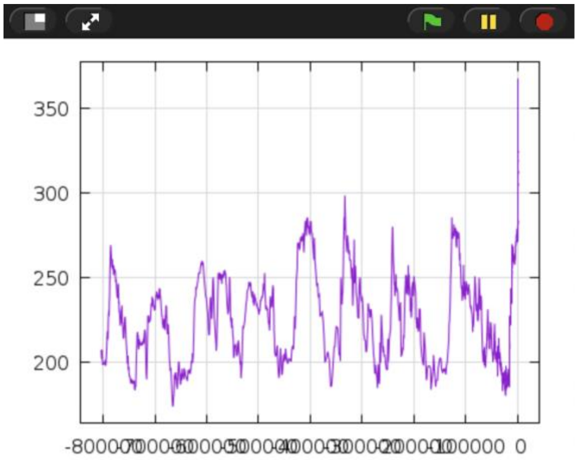
call IceCoreData / getCarbonDioxideData Antarctic Composite startyear endyear

1901	A	B
1	-803718.87	207.29
2	-803182.44	202.23
3	-802572.67	204.86
4	-802059.87	207.5
5	-801975.28	202.92
6	-801759.25	202.65
7	-800922.09	198.66
8	-799286.06	198.68
9	-796731.3	200.49
10	-795428.82	197.91
11	-794902.59	198.79
12	-793737.32	204.62
13	-793172.45	208.79
14	-791902.76	218.39
15	-791569.39	213.36
16	-790667.3	214.45
17	-790080.78	224.69
18	-789106.88	230.55
19	-788689.41	227.38
20	-788003.01	235.49
21	-787544.04	238.64
22	-786741.87	257.02
23	-786529.24	257.2
24	-785853.57	266
25	-785405.51	269.36
26	-784613.26	264.35
27	-784239.6	262.23
28	-783569.05	259.66
29	-783084.08	255.85
30	-782311.76	260.92
31	-781248.53	255.02
32	-780778.45	257.44
33	-779936.68	251.51
34	-779612.83	257.21
35	-778769.44	252.52
36	-778238.92	254.43

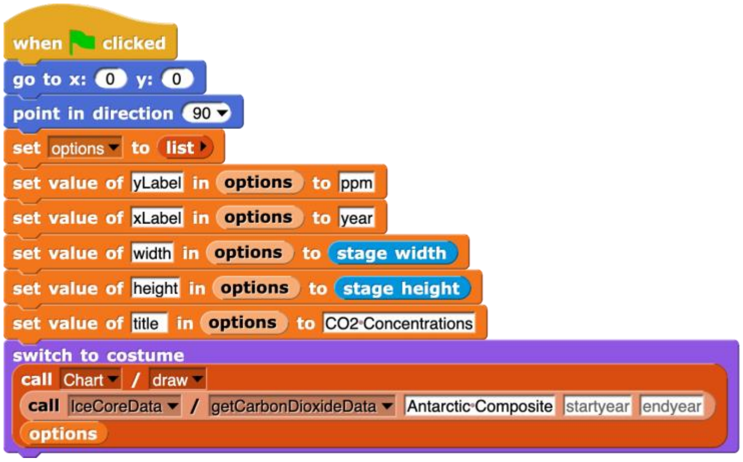
And we also know how to plot it quickly:



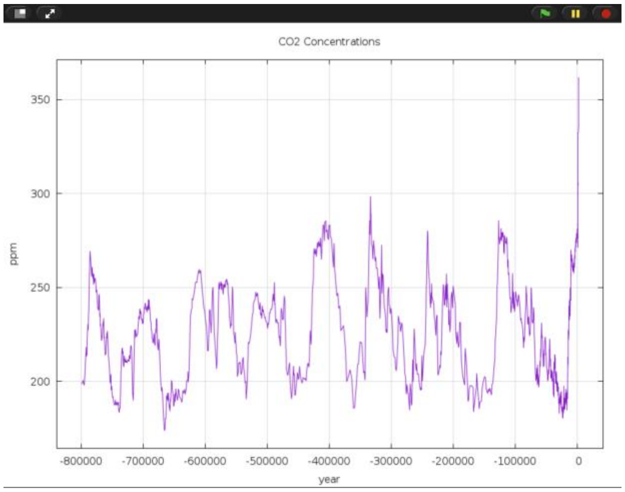
```
switch to costume
  call Chart / draw
    call IceCoreData / getCarbonDioxideData Antarctic Composite startyear endyear
    options
```



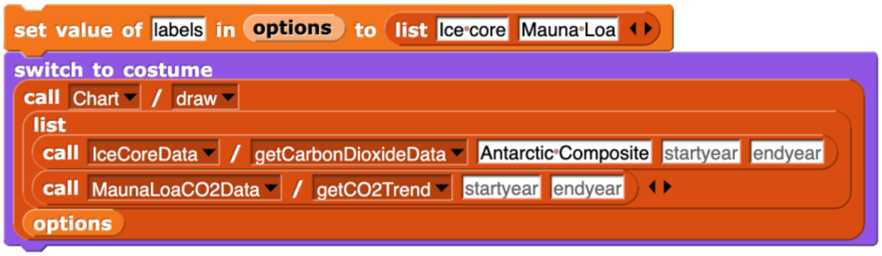
Let’s make this plot prettier first of all, but also interactive, so that we can look at various parts of the data instead of just the whole range. First, I am going to increase the stage size to give more room for the labels to avoid overlap. You can do that by clicking the Cogwheel button on top and selecting “Stage size.” If you have room, double the default size to 960 by 720, but you can set it to whatever size you prefer. Once we do this, we need to adjust the options for the **`draw`** RPC. Let’s do that and add a title and some labels:



```
when ⚑ clicked
go to x: 0 y: 0
point in direction 90
set options to list
set value of yLabel in options to ppm
set value of xLabel in options to year
set value of width in options to stage width
set value of height in options to stage height
set value of title in options to CO2 Concentrations
switch to costume
  call Chart / draw
    call IceCoreData / getCarbonDioxideData Antarctic Composite startyear endyear
    options
```

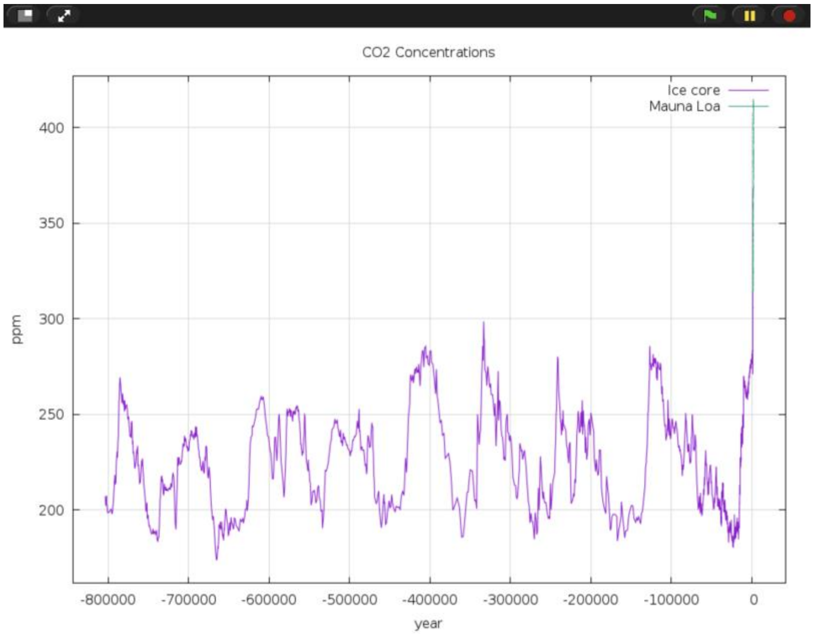


We know how to include multiple plots in the same chart, so why don't we add the Mauna Loa data for comparison? And we can also add a legend with the **`labels`** option:

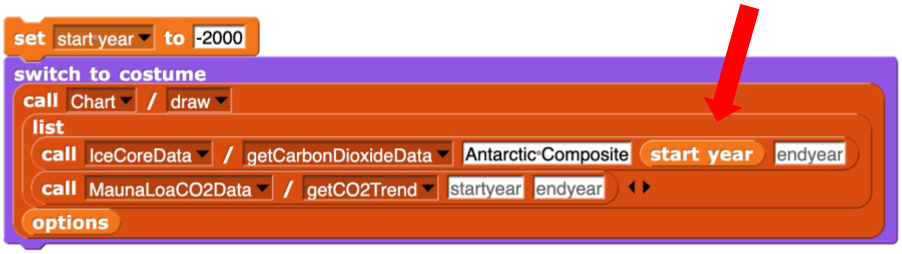


```
set value of labels in options to list Ice core Mauna Loa ◂▸
switch to costume
  call Chart / draw
    list
      call IceCoreData / getCarbonDioxideData Antarctic Composite startyear endyear
      call MaunaLoaCO2Data / getCO2Trend startyear endyear ◂▸
    options
```

We replaced the input argument **`lines`** of the **`draw`** RPC to have a list of two data arrays: one for the ice core and one from Mauna Loa. The chart service takes care of the rest:

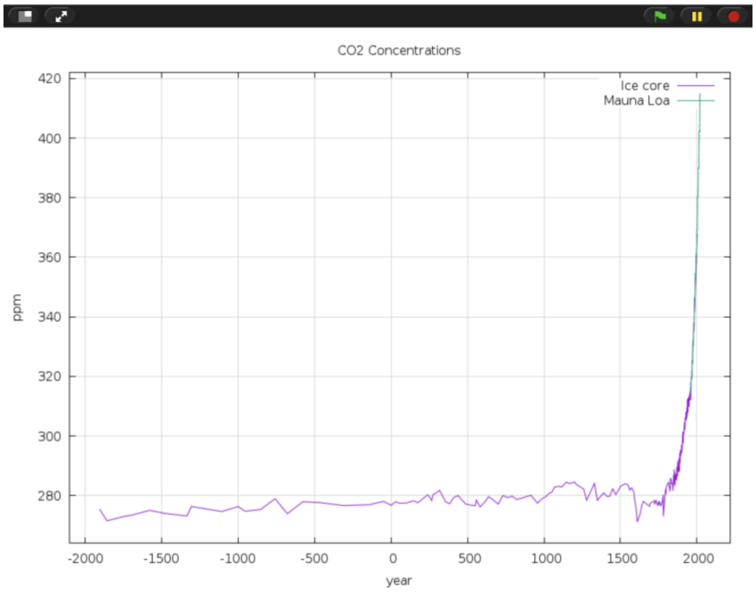


Check out the green plot: an almost vertical line. It would be interesting to be able to inspect that region closer, right? Let's introduce a variable called **`start year`** that we will adjust to see different time regions:



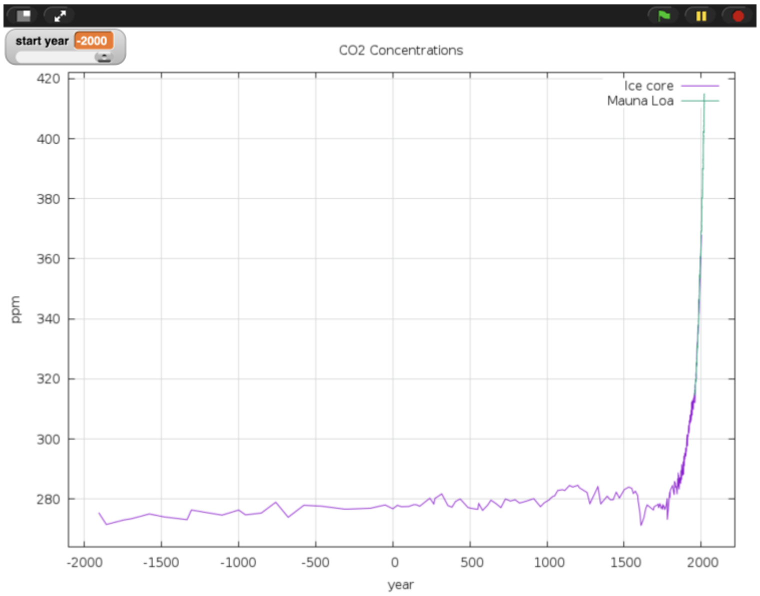
```
set start year to -2000
switch to costume
  call Chart / draw
    list
      call IceCoreData / getCarbonDioxideData Antarctic Composite (start year) endyear
      call MaunaLoaCO2Data / getCO2Trend startyear endyear ◂▸
    options
```

We set it to -2000 and include it in the **`getCarbonDioxideData`** RPC as an input. This is what we get:



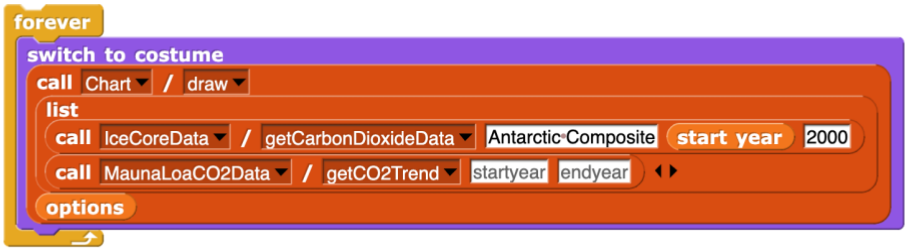
We are looking at the past 4000 years now. And the green line is still almost vertical. Interesting…

Let’s make **`start year`** user-adjustable. We can click the checkbox next to its name in the Variables tab to make it visible on the Stage. If you right click on it, you can make it a slider. But we also need to make sure that the minimum and maximum values for it are what we need. Again, we can specify these by right clicking on the variable in the Stage and selecting “slider min” and “slider max.” I typed in -800000 for the minimum value and 1800 for the maximum. Now we can slide the slider and we are changing the variable. Very nice!



But wait! The plot is not changing at all! What is going on?

Of course, the plot is not changing. We use this **start year** variable only once. We create the chart and display it and our script is done. It does not care if we change the variable at all. What we need to do is put the plotting code into a loop:

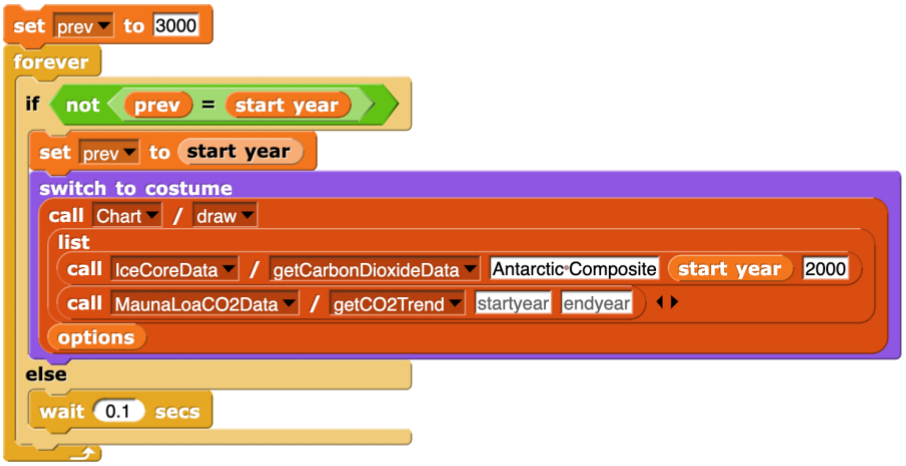


```
forever
  switch to costume
    call Chart / draw
      list
        call IceCoreData / getCarbonDioxideData Antarctic Composite start year 2000
        call MaunaLoaCO2Data / getCO2Trend startyear endyear ◂▸
      options
```

While this works, this is super wasteful. It keeps calling the **draw** RPC continuously at max speed when most of the time the **start year** variable is not changing at all. We should do two things:

1. Check whether the **start year** variable has changed and only call the **draw** RPC if it has,
2. Wait a fraction of a second before checking again.

Here is our final version:



```
set prev to 3000
forever
  if not prev = start year
    set prev to start year
    switch to costume
      call Chart / draw
        list
          call IceCoreData / getCarbonDioxideData Antarctic Composite start year 2000
          call MaunaLoaCO2Data / getCO2Trend startyear endyear ◂▸
        options
  else
    wait 0.1 secs
```

We introduced a new variable called **prev**. This will keep track of the **start year** value the last time we noticed that it changed. We initialize it to a value that **start year** can never be (3000 in this case). Then in the forever loop, we check whether **prev** and **start year** are the same or not. If they are the same, we do not need to do anything, so we just wait one tenth of a second before checking again. If they are not the same, we need to update the chart. First, we update the value of **prev**. It is actually important that we do this first. Since the **draw** RPC may take some time (a couple of tenths of a second) and in the meantime, the user might be sliding the slider. If we set **prev** after the RPC call, we may miss the change the user made in the meantime. Finally, we call the **draw** RPC to refresh the chart.

Here is the final version:
https://editor.netsblox.org/?action=present&Username=ledeczi&ProjectName=IceCoreDemo&

As an exercise, you can add a **stop year** variable and zoom in any of the regions that way. However, care must be taken that **stop year** remains greater than **start year** at all times.