Lesson 5 - Logic & Loops

Function: To check conditions and determine behaviour of the program

Permafrost

Permafrost consists of any frozen ground that has been frozen (below 32°F/ 0°C) permanently (did not thaw) for two years minimum and can remain frozen for thousands of years. Rising global temperatures however, have caused permafrost to thaw at unprecedented rates, which is problematic for a variety of reasons. First, the decomposition of organic matter is paused with the freezing of the ground in a similar fashion to how a freezer pauses the decomposition of food. This means that as the ground thaws, organic matter that was frozen begins to decompose, releasing large quantities of greenhouse gases. Methane, which is significantly more potent than carbon dioxide, is currently being released at rates of 17 million tonnes per year due to melting permafrost.

Permafrost is also a massive reserve of carbon - approximately 1,500 billion tonnes. It is estimated that regions such as the Arctic could potentially release 2 times the amount of carbon that currently exists, and [one study](#) predicts that a total of 92 billion tons of carbon could be released from now until 2100 due to melting permafrost. Another shocking [discovery](#) that has serious implications on ecosystem and human health, is the significant amount of mercury found in permafrost - a whopping 15 million gallons (approximately 56,781.18 tonnes). The release of greenhouse gases is not the only effect of melting permafrost. Permafrost regions which house approximately 35 million people, including many regions in the Arctic, a region that is melting far faster than predicted, face major infrastructure damages such as [collapsing buildings](#) and [shifting roads](#), and therefore also face [significant costs](#). [One study](#) estimates that a total of 22,000 km of road, 8,800 km (40%) of which is located in Canada, 4,400 km (20%) in the Russian North, 4,400 km (20%) in Alaska, and another 4,400 km (20%) in the Nordic Region.

To read more about permafrost, visit the [National Geographic online encyclopedia](#). Imagery is courtesy of [the NASA Earth observatory](#).

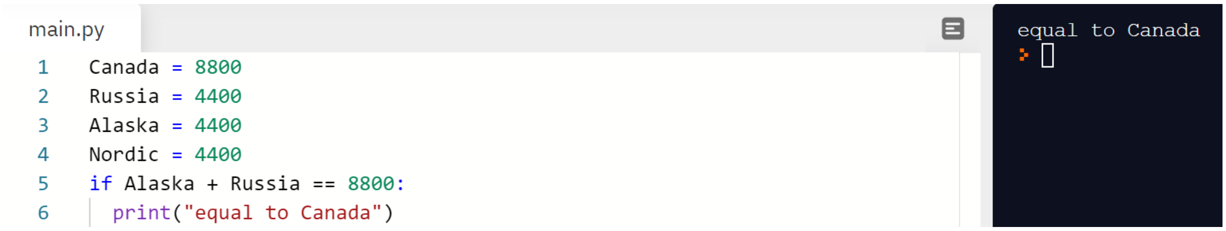
Part 1: Create an if statement using comparison operators

1. Create the variables for each country (Canada, Russia, Alaska, and Nordic region) and the number of kilometers impacted by thawing permafrost.
2. Use if followed by a logic expression to add the data for Alaska and Russia.
3. To check if these variables are equal to 8800 km (rather to assign a value or define a variable) use two equal signs ==
4. End the if statement with a colon :

   if Alaska + Russia == 8800

5. Indent the next line and print “equal to Canada”

Example:



```
Canada = 8800
Russia = 4400
Alaska = 4400
Nordic = 4400
if Alaska + Russia == 8800:
  print("equal to Canada")
```

```
equal to Canada
```

Part 2: Create an else statement

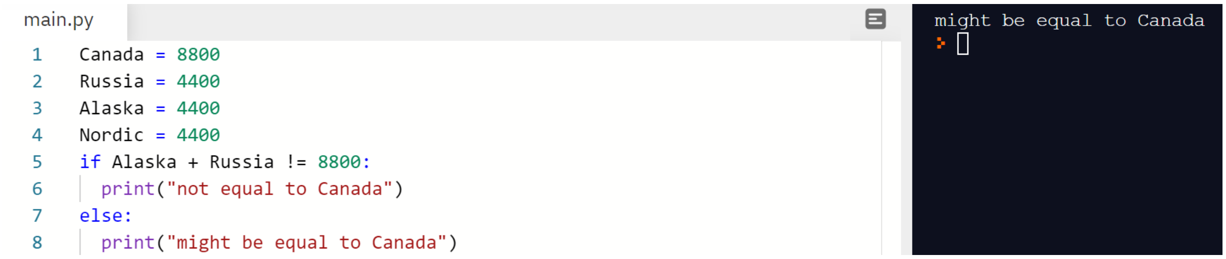
1. Using an if statement, assess whether adding Alaska and Russia does not equal to 8,800 km of damages road infrastructure by using an exclamation mark and an equal sign !=

   if Alaska + Russia != 8800:

   print(“not equal to Canada”)

2. Since this condition is false, to execute a different statement, use the else statement by typing else on a new line, followed by a colon :
3. Indent the following line and print “might be equal to Canada”

Example:



```
Canada = 8800
Russia = 4400
Alaska = 4400
Nordic = 4400
if Alaska + Russia != 8800:
  print("not equal to Canada")
else:
  print("might be equal to Canada")
```

```
might be equal to Canada
```

In this example, if Alaska and Russia do not equal 8800 km, then a statement “not equal to Canada” would have been printed. However, since that condition is false, the program will resort to the default statement “might be equal to Canada”.

So far the comparison operators equal to == and not equal to != have been used to create if statements. This is a list of commonly used comparison operators in python:

Operator	**Example**	**Meaning**
==	a==b	The value of a is equal to the value of b
!=	a != b	a is not equal to b
<	a < b	a is less than b
<=	a <= b	a is less than or equal to b
>	a > b	a is greater than b
>=	a >= b	a is greater than or equal to b

Part 3: Create an elif statement

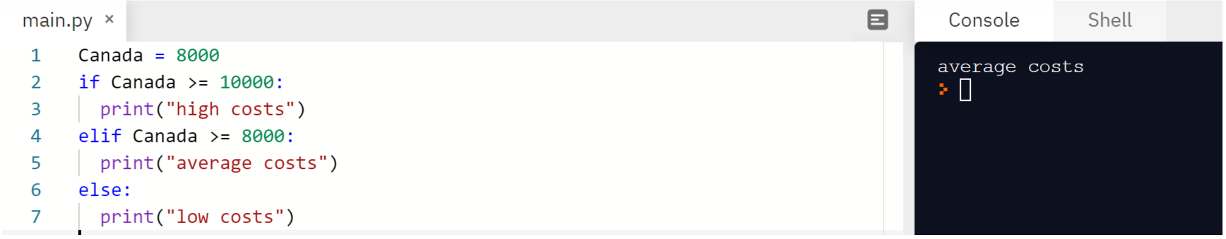
1. Using the same variables as parts 1 and 2, create an if statement to determine whether Canada is equal or greater than >= 10,000 km and should that condition be true, to print “high costs”.

    ```
    if Canada >= 10000:
        print(“high costs”)
    ```

2. To create a chain of conditions (in this case, a second condition), create an elif statement using the same syntax as the if statement, except replace the word if with elif (combination of else and if). For this elif statement, if Canada road damage is equal or greater than 8000 km, then print “average costs costs”.

3. Create an else statement to indicate that otherwise, the statement “low costs” should be printed should the first two conditions not be true.

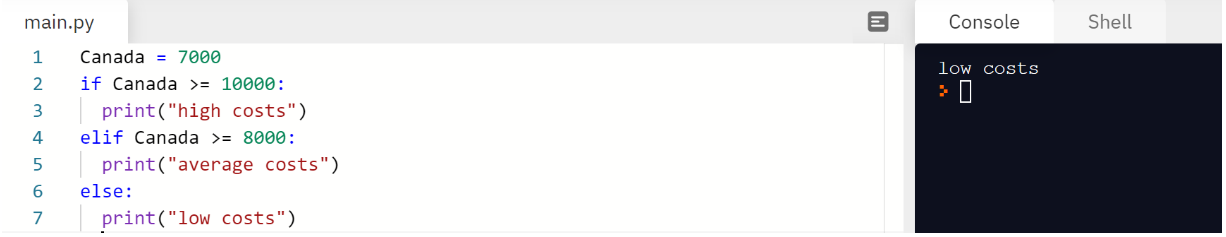
Example:



```python
Canada = 8000
if Canada >= 10000:
    print("high costs")
elif Canada >= 8000:
    print("average costs")
else:
    print("low costs")
```

```
average costs
```

In this example, the if statement is not true, but the elif statement is true. However, if the variable for Canada were to change to 7000 km instead of 8800 km, then the elif statement would be false, defaulting to the else statement and “low costs” would be printed instead.



```python
Canada = 7000
if Canada >= 10000:
    print("high costs")
elif Canada >= 8000:
    print("average costs")
else:
    print("low costs")
```

```
low costs
```

Many elif statements can be created but only one else statement can be created, though they are optional.

✅***Task 1: To assess whether land has been frozen long enough in order to be considered permafrost, create if, elif, and else statements, that print “not frozen long enough”, “frozen just long enough”, and “permafrost” for the variable of time of 1.2 years.***

✅***Task 2: repeat the same logic if time is equal to 2 years and 1000 years.***

Part 4: Write a conditional expression using different operators

1. Using the following integers from the text (permafrost thawing is estimated to release **17** million tonnes of methane per year, and hold a total of **1500** billion tonnes of carbon, **92** billion tonnes of which are to be released from now to the year 2100) create the variables thawing_1, thawing_2, and thawing_3 respectively.

2. This is a list of commonly used operators in python. For this exercise, use the modulus operator % to divide the variables and determine the remainder.

Operator	Example	Meaning
+	a + b	addition
-	a - b	subtraction
*	a * b	multiplication
/	a / b	division - result can be a float
//	a // b	floor division - a divided by b, rounded down to the smallest whole number
**	a ** b	exponentiation - a to the power of b
%	a % b	modulus - the remainder when a is divided by b

3. Using an if, elif and else statement and a modulus operator % (which determines the remainder after division), print the string “permafrost” if the thawing_1 is divisible by 2 and print the string “is thawing” if thawing_1 is divisible by 5 (implying that after division, the value is equal to == 0). Otherwise, have the program print the number.

```
thawing_1 = 17
if thawing_1 % 2 == 0:
        print(“permafrost”)
elif thawing_1 % 5 ==0:
        print(“is thawing”)
else:
        print(thawing_1)
```

As the integer 17 is not divisible by 2 nor 5, the output should be 17.

✅***Task 3: create the same conditional expression for thawing_2 and thawing_3.***

✅***Task 4: using the same logic, add an if statement that prints “permafrost is thawing” if the variable is both divisible by 2 and 5. Do this for the variables thawing_2 and thawing_3.***

Part 5: Create a while loop

1. Since permafrost is land that has been frozen for at least 2 years and up to thousands of years, create a variable for permafrost that is equal to 2.

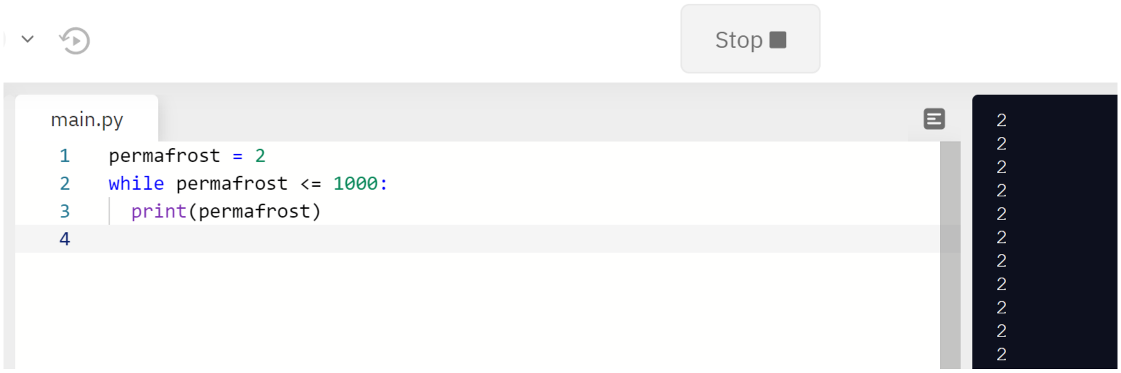
    permafrost = 2

2. Use a while loop. This loop states that while a condition is true, another statement can be executed and will continue until the condition is false. Using the same syntax as previous loops, write a while statement that states that while the variable is less than or equal to 1000, the variable needs to be printed.

    ```
    while permafrost <= 1000:
        print(permafrost)
    ```

3. Since the value assigned to the variable is 2 and the condition is still true (it is less than or equal to 1000, the number 2 will continually be printed (also known as an **indefinite iteration** in python). You can stop it by manually pressing the stop button.

Example:



```
permafrost = 2
while permafrost <= 1000:
  print(permafrost)

```

```
2
2
2
2
2
2
2
2
2
2
2
```

✅***Task 5: Using this example, add another statement to print all numbers from 2 to 1000 incrementally.***

Part 6: Create a for loop

A **for** loop is used to specify items within certain parameters, also called a definite iteration in python.

1. Create a list for permafrost that includes the years 2,3,4,5.

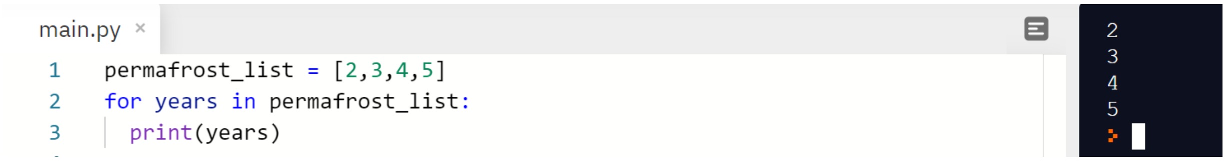
   permafrost_list = [2,3,4,5]

2. Use the syntax for x in list: (x representing whatever variable you want to select from the list - in this case it is the years in the list). The variable years is introduced in this for statement.

   for years in permafrost_list:

3. Print the variable years (this should print the whole list).

Example:



```
permafrost_list = [2,3,4,5]
for years in permafrost_list:
  print(years)
```

```
2
3
4
5
```

Part 7: Create a for loop using range

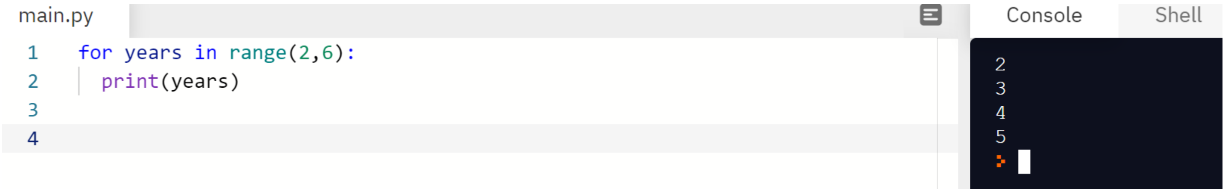
Instead of using a list, the range function can return numbers in sequence, ending before the number that is specified.

1. Use a range function to only print years 2,3,4,5. Use the same for statement as in the previous example, but instead of asking for years in permafrost_list, ask for years in range(start, stop). To select only years 2,3,4,5 but not 6, select the number to start the sequence (otherwise the default will be zero), followed by a comma and the number before which you would like the sequence to stop.

   for years in range(2,6):

2. Print the variable years.

Example:



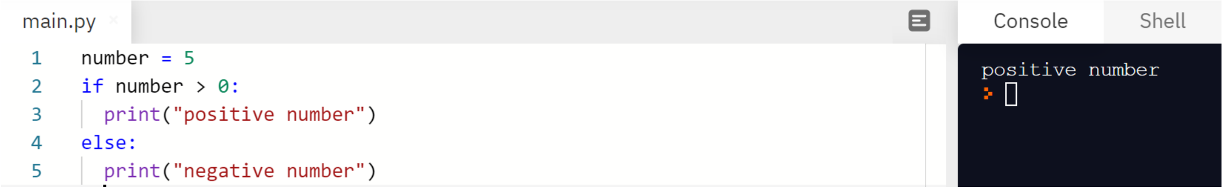
```
for years in range(2,6):
  print(years)


```

```
2
3
4
5
```

Part 8: Create a for loop with an if statement

1. Create an if/elif/else statement that prints “positive number” or “negative number” for the numbers 5, -3, 7, and -10.

Example:

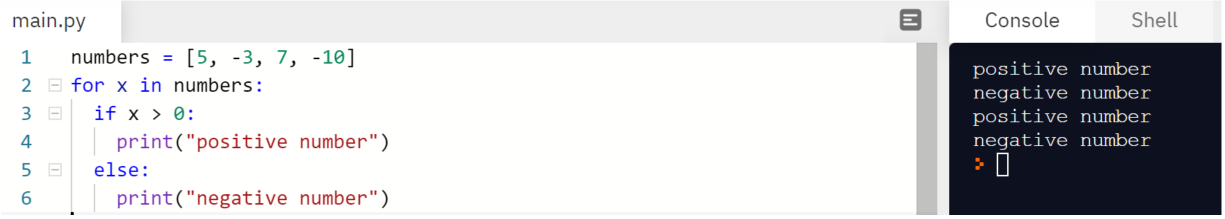


```python
number = 5
if number > 0:
  print("positive number")
else:
  print("negative number")
```

```
positive number
>
```

2. However, using an if statement requires you to input each value independently. To cycle through the whole list, use a for loop and embed the if statement within it.

Example:



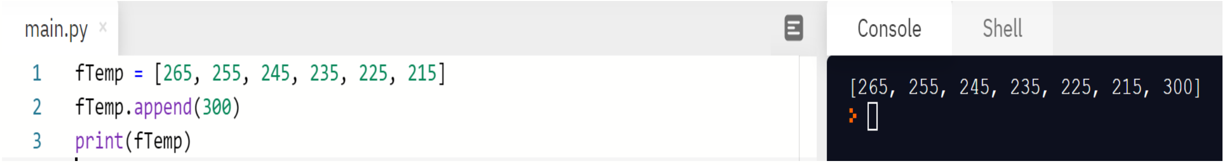
```python
numbers = [5, -3, 7, -10]
for x in numbers:
  if x > 0:
    print("positive number")
  else:
    print("negative number")
```

```
positive number
negative number
positive number
negative number
>
```

Part 9: Appending lists in a for loop

So far we have learned how to add a value to an existing list using the append function.

For example:
This is a list containing temperatures in fahrenheit. To add another value at the end of the list, you can append the list:



```
fTemp = [265, 255, 245, 235, 225, 215]
fTemp.append(300)
print(fTemp)
```

```
[265, 255, 245, 235, 225, 215, 300]
```

What if you wanted to create a new list and convert the values to celsius?

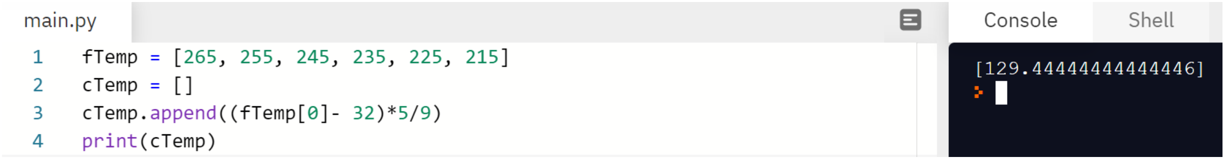
1. Create a new list that is empty. It is empty because you will add a new value to it later that converts the temperature in fahrenheit to celsius.
   cTemp = [ ]

2. To do this with only one value from the list, the first value, append the new list to add new values. Instead of inserting a value into the parentheses (like the previous example), insert the list that you want to convert, fTemp, followed by an index to reference the first value, followed by the formula to convert fahrenheit to celsius (x - 32)*5/9

   cTemp.append((fTemp[0]- 32)*5/9)

3. Print the new list.

Example:



```
fTemp = [265, 255, 245, 235, 225, 215]
cTemp = []
cTemp.append((fTemp[0]- 32)*5/9)
print(cTemp)
```

```
[129.44444444444446]
```

As you can see in this example, this only allows you to convert one value at a time. To convert all the values at the same time, you would need to use a for loop.

✅***Task 6: create a for loop to convert the entire list of temperatures in fahrenheit into celsius.***