Lesson 4 - Learning to Speak Binary and Parseltongue

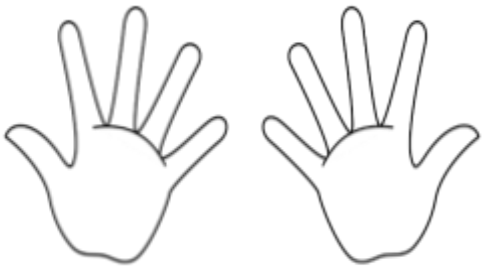
New Concepts

- Computers can do math… who knew?!
- Activity: make a program to calculate your age in dog years

Computers and Math

Now we know that computers like to use NUMBERS. Since a computer is made of millions and millions of switches, computers think in kind of funny numbers.

How do we humans count? From 1 to…… 10? Right! Funny thing, how many fingers do humans have?



So, computers don't have hands, but they do have SWITCHES! Switches don't have fingers, but they do have positions… ON and OFF

Computers only count to…. 1! We call this “base 2” or Binary!

So, how does a computer count to 10? It uses a system of Bits and Bytes. Bits are the smallest unit of data that a computer can hold. When we put 8 of them together we get a byte! By assigning each bit a different value and doing some basic addition, we can use fewer bits to store more numbers.

The first bit is 1. It lets us count to 1. To count to 2, we need a second bit - 2. But now we can make three by adding 1 and 2 together. To keep going, we need another bit - 4. Then we can use addition to get 5, 6, and 7, but need to add another bit to get to 8.

So, the first four bits are 8, 4, 2, 1. In the 4 blanks by each number, put a 0 if you don’t need that number and a 1 for those that you do. For example, I don’t need any of those numbers to make 0, so I would write 0 0 0 0. I only need the 1 to make 1, so I write 0 0 0 1. 2 is easy as well, but how do I write 3? See if you can figure out all the numbers up to 10.

8 4 2 1

1 > __ __ __ __	6 > __ __ __ __
2 > __ __ __ __	7 > __ __ __ __
3 > __ __ __ __	8 > __ __ __ __
4 > __ __ __ __	9 > __ __ __ __
5 > __ __ __ __	10 > __ __ __ __

Kind of funny, but It works! Can you figure out what number the next bit stands for?

Making a Computer Do Math

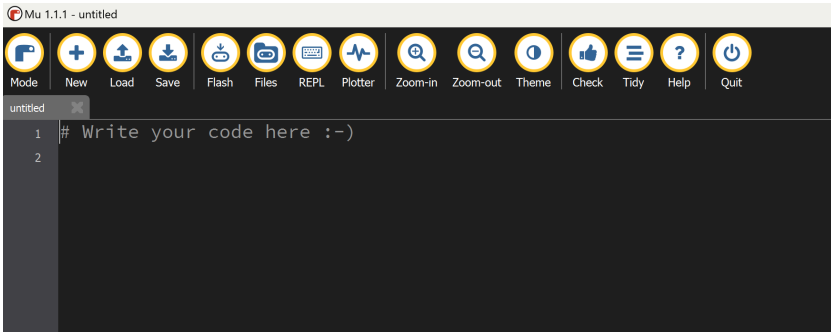
While 1’s and 0’s are cool, it’s kinda hard to keep straight. Luckily, we made computer languages that let us tell the computer what we want it to do. This week we are using a language named after a snake…

In Python, we can write code to let the computer do math for us!

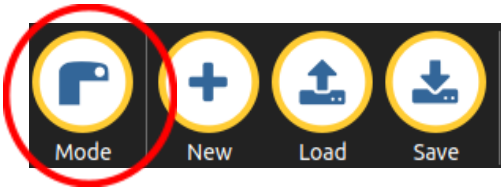
Activity 1: Dog Years!

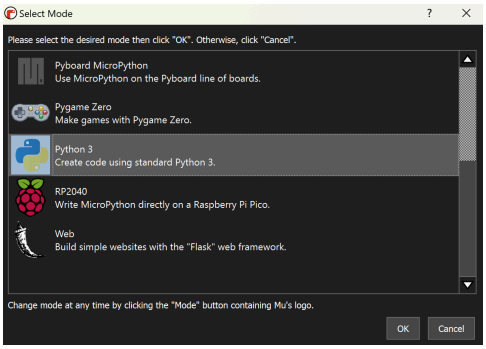
In this activity, we will be creating a python program that calculates your age in dog years for you! It will ask you what your age is in human years, then do all the math to tell you how old you are in dog years! Pretty neat, eh?

On your computer, open `Mu` (sometimes known as `Mu Editor`). This is our `INTEGRATED DEVELOPMENT ENVIRONMENT`, or more commonly known as an `IDE`. This is a fancy computer/engineer language that essentially means “an application to program/write code in”. When it finishes launching, you should land in a window that looks like this.



Let’s make sure you are in the right mode - in the top left, click the `Mode` button and select `Python3` in the pop-up window.

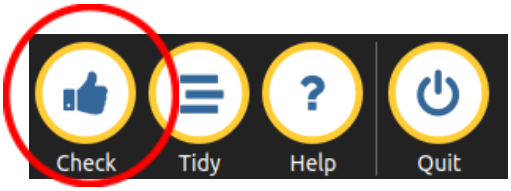


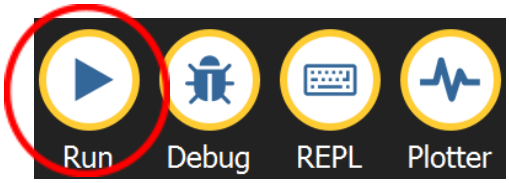


Now, type the following code. **A word of warning!** Computers are great at doing exactly what you ask them to do. If the letter in the example program is capitalized or lower case, it must be the same way when you type it. Also, all of your punctuation (especially the colons!) have to be there for the computer to be able to understand what you are telling it to do. And pay special attention to spacing - python is a ‘whitespace sensitive’ programming language, which means that it cares a lot about how code is spaced.

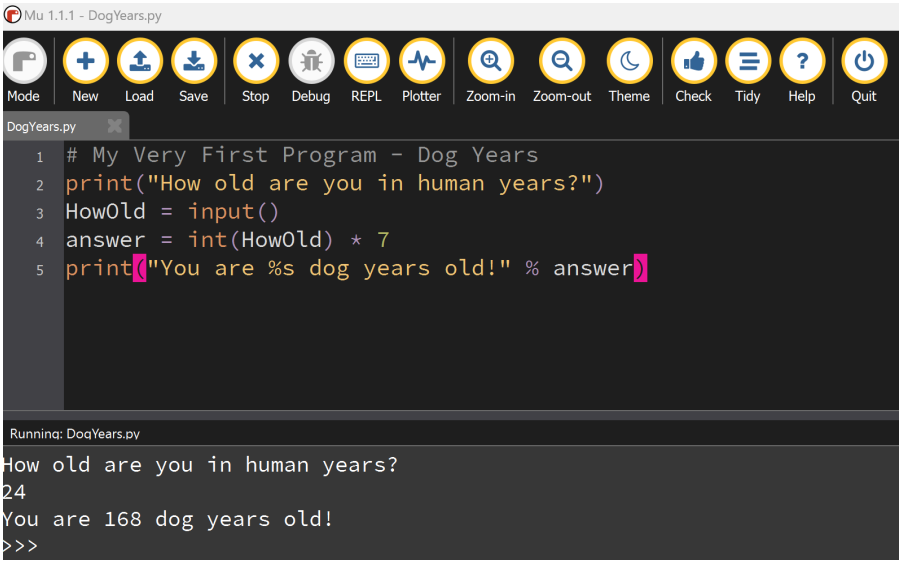
```
# My Very First Program - Dog Years
print("How old are you in human years?")
HowOld = input()
answer = int(HowOld) * 7
print("You are %s dog years old!" % answer)
```

To run your program, click the Check button (the thumbs up) to check your code for any errors. If you have a few errors, fix them up and use the Check button until you have no more errors. When there are no more errors, you are good to go!





Now, click the Run button! A window should pop up, asking you to name your program. Name it something like DogYears.py. After saving, a terminal should appear and ask you for your age! Go ahead and type it in, then press enter!



Comprehension Check

- What is #?
- What is HowOld?
- How did the computer do our math for us, if it can only use 1's and 0's?

Activity 2: Data Conditions

Our first program was a huge success! But, what if we wanted to do something based on someone's age? For example, if they are greater than 100 years old, we could call them old! Let's use an *if/else* statement in python!

Simply put, an `if/else` conditional statement checks `if statement a` is true. `If` it is, then we do something, `else` we don't do anything at all.

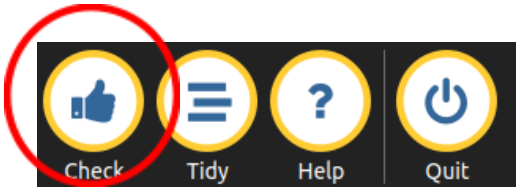
Back in Mu, let's revise our code to call someone old if they are greater than 100 years old, otherwise calculate their dog age.

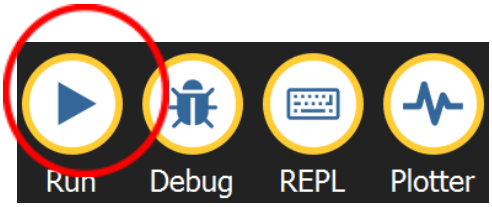
```
# My Very First Program - Dog Years
print("How old are you in human years?")
HowOld = input()

if int(HowOld) > 100:
        print("You are really old!")
else:
        answer = int(HowOld) * 7
        print("You are %s dog years old!" % answer)
```

To run your program, click the `Check` button (the thumbs up) to check your code for any errors. If you have a few errors, fix them up and use the `Check` button until you have no more errors. When there are no more errors, you are good to go!

Now, click the `Run` button! A terminal should appear and ask you for your age! Go ahead and type it in, then press enter!





Challenge: Change it up! Calculate age in horse years, chicken years, dragon years, etc. or calculate something else all together! You could calculate how many individual eggs you have based on how many egg cartons you bought. Use what you've learned about **variables** and **if/else statements** to make something fun.