# Laboratory 2

September 3 and 5, 2019

Strings again:

Python strings are delimited by the single or the double quotation mark. If you use three double quotation marks as delimiters before and after the string, then the string can contain line breaks. In this exercise, we just use several print statements to print out one line at a time.

Recall that a backslash followed by many other characters has a special meaning. We say that the backslash is an *escape character* and that the symbol itself has been escaped. If you want to put a quotation mark inside a string delimited by the same quotation mark, then you need to use the backslash. While Python is very good at divining your intention (or rather has a clever algorithm to understand strings), you sometimes need to escape the backslash itself in order for it to be represented. As a rule of thumb, when you want a backslash, always escape it by doubling it.

1   Your first task is to draw the bear's face on the right with seven print statements. (The long horizontal lines are underscores.) The second print statement is something like
`print(" / \\.-\"\"\"-./ \\")`
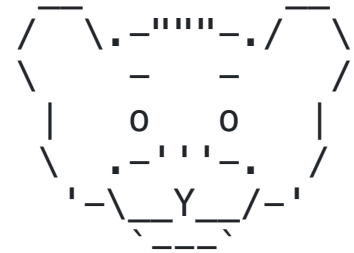
Using the input function:

2. Recall that Python uses the input function in order to obtain input from the keyboard. Using the following Furlong to Meters program as a paradigm, **create programs** that convert Cables to Meters, Acres to Square Meters and Gallons to Liters.

```
furlongs = float(input("Enter distance in Furlongs: "))
print('The distance is', furlongs*201.168, 'meters.')
```

Notice the following two points: The parameter of the input function is the prompt string. It looks better with white space at the end, either a space or a tab character. Second, the return value of the input function is always a string that needs to be cast with int or float.

3. Write a program that asks the user for the x and y coordinates of a point. The program then prints out the distance $\sqrt{x^2 + y^2}$ of the point from the origin.

Algorithms

4. Assume that you want to solve a complicated equation such as $x^5 - x^4 + 2x^3 - 2 = 0$. The bisection method is a very simple algorithm that works as long as you can calculate the expression on the left with reasonable accuracy. We observe that
$$(-2)^5 - (-2)^4 + 2(-2)^3 - 2 < 0$$
and that
$$2^5 - 2^4 + 2 \cdot 2^3 - 2 > 0.$$
Calculus tells us that there is a solution in the interval (-2, 2) because the function value is negative at one endpoint and positive at the other. We then **bisect** the interval, obtaining the middle point 0. We evaluate the function at 0 and obtain $0^5 - 0^4 + 2 \cdot 0^3 - 2 = -2$, which

is negative. Therefore, there must be a solution of the original equation between 0 and 2. We next try out 1, and find that 1 is indeed a zero.

Now we want to solve the slightly different equation $x^5 - x^4 + 2x^3 - 3 = 0$. By evaluating the left side at -2 and 2, we again get to know that there is a solution between -2 and 2. Currently, we cannot yet write a program that implements the bisection method, but we can write a program that asks the user to enter a value for x and that prints out the value of $x^5 - x^4 + 2x^3 - 3$. With the help of this program, we can apply the bisection method in order to obtain an interval of length less than 0.1 in which there is a solution to this equation.

Soon, we will implement the bisection method completely.

5. The Raphson-Newton method for the calculation of the fourth square root of a number a starts with a random guess $x = 1$ and repeatedly improves the guess by calculating

$$x - \frac{x^4 - a}{4x^3}.$$

Write a program that asks the user to input a, and that then prints out the first 10 guesses for $\sqrt[4]{a}$. Try out your program with small numbers (such as 5 or 7) and with large numbers such as 501, 5001, …). As you will see, we have too many iterations for small numbers and not enough for larger numbers.

6. The method can be easily extended to larger roots. The n-th root of a can be obtained by first guessing $x = 1$ and then repeatedly replacing x by

$$x - \frac{x^n - a}{n x^{n-1}}.$$

Write a program that asks the user for the value of a and n and then calculates the repeatedly improved guesses 10 times.