# Project CSGirls Knoxville Coding Workshop
----------------------------------------------------------------

## What is a computer program?

A computer program is a collection of instructions that performs a specific task when executed by a computer. Computer programs follow a series of steps, known as an **algorithm.** Using an algorithm is like following a recipe, because both describe a series of steps to follow that produce a result. Thus, a computer algorithm is a procedure that allows a computer to solve a problem.

## What is Python?

Python is a type of computer programming language that allows the user (you) to specify instructions that the computer can understand and follow. Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code.

This workshop uses the Python browser coding environment, *codeskulptor.org.*

A browser coding environment means that you can type the link, and start coding!

## Printing to Console

The first piece of code we will write will print some text to the console. The print() function is used to print a String to the console for the user to read and interpret. For example, the following code will print "Hello World!" to the output console (try it out yourself!).

**Ensure that there are no tabs or spaces before the line of code. Python reads the white spaces to execute the line of code (see end of lesson for more details).**

print "Hello World!"

Congratulations! You have written your first Python program.

## Printing Variable to Console

We can also print the contents of a variable to the output console similar to how we have done before. Try out the examples below!

```
myNumber = 5
print myNumber

myString = "Hello World!"
print myString

name = "John"
print "Hello, my name is" + name
```

## Mathematical Operations

In Python, we can perform numerical operations on **int** (or **float** ) type variables and store the result in a variable. Try out the examples below and **print the results to the output console** (just like we did before).

```
val = 0
val += 10

val = 60
val -= 5

val = 2
val *= 2

val = 1000
val /= 2
```

Can you guess what mathematical operation took place in each case?

## Variable Type Conversion

In Python we can declare a string, int and float variable (plus a few more we haven't covered yet) and later convert them to a different variable type.

```
myString = "100" myInt = ( int )(myString)

myInt = 42
```

```
myFloat = ( float )(myInt)

myFloat = 18.4
myString = ( str )(myFloat)
```

Try the examples above and **print the result to the console** to see what operation takes place in each case.

## Console Input

Rather than hard-code the string we are printing to the output console, we can also input a string to the console by executing the script and entering keyboard commands.

```
message = input( " " ) print message
```

In the above piece of code, a string that contains whatever the user types will be stored in the message variable for later use.

We can also do something like the following:

```
name = input( "What is your name?\n" ) # user inputs their name
age = input( "What is your age?\n" ) # user inputs their age
location = input( "Where do you live?\n" ) # user inputs where they live

print "My name is " + name print "I am " + age + " years old" print "I live in " + location
```

## Number Guessing Game Project

Below is a block of code that will create a number guessing game. The game will generate a random number at the beginning and the user will input a number that they think may be the same as the randomly generated number. If the user guesses right, then "You Won!" will be printed to the console. If the number the user guessed is lower than the random value, then "Higher!" will be printed to the console and similarly if the guessed number is higher than the random value, "Lower!" will be printed to the console.

To make things exciting, some of the code has been excluded from the block below. It is up to you to **fill in the blanks** to complete the code so the game works as its supposed to using all you have learned in this lesson up to now.

**Hint:** the lines beginning with "#" are comments that do not affect the code at all. They are there to help you and tell you what each line of code should do.

```python
import random # import random library to generate random values

randomNumber = random .randint(1, 10) # generate random number between 1 and 10

guessedNum = _____ # user inputs an INT that will be compared with the random value

if guessedNum == randomNumber: # if guessed number correct, print "You Won!" _____

elif guessedNum < randomNumber: # if guessed number less than random number, print
"Higher!" _____

elif guessedNum > randomNumber: # if guessed number greater than random number, print
"Lower!" _____

while guessedNum != randomNumber: # Above code repeats until guessed number correct
_____ # Same code as above (don't import random library or generate random number
again)
```