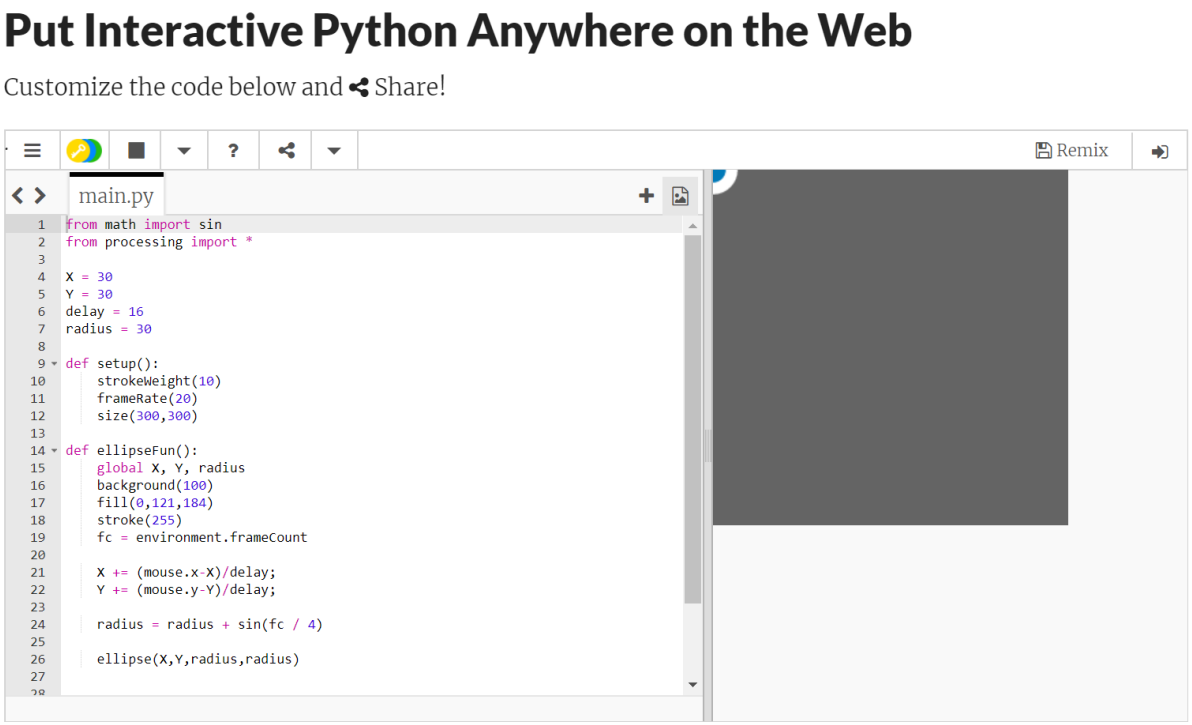
Lesson 1

A very brief history of Python

- Python is a interpreted high-level programming language for general-purpose programming.
- Created by Guido van Rossum and first released in 1991.
- Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code.

Trinket

Trinket is an entirely in-browser implementation of Python. It is a helpful tool to run Python code as it allows user input to the output console.



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)
```

Note: If you are ever confused about what type of variable you are dealing with (string, int, float etc.) you can use the type() method and print out the result to see the variable type.

```
myVariable = 5

print(type(myVariable)) # prints out "<type 'int'>"
```

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 example below and **print the results to the output console** (just like we did before).

```
val = 10

val += 5
```

Tasks:

1. Replace the code above with a **subtraction** operation (**-=**)
2. Replace the code above with a **multiplication** operation (***=**).
3. Replace the code above with a **division** operation (**/=**)
4. Replace the code above with a **modulo** operation (**%=**).

Use different values for each task!

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)
```

Note: "\n" is used in a String when you wish to start a new line. The print function does this automatically, which is why we don't have to use it each time we print.

Tasks:

1. Prompt the user to input their age.
2. Prompt the user to input where they live.
3. Print out a sentence that says the age of the user.
4. Print out a sentence that says where the user lives.

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 “Guess Higher!” will be printed to the console and similarly if the guessed number is higher than the random value, “Guess Lower!” will be printed to the console.

Also, don’t worry if you have not seen some of the code before. We will cover it all in the near future but it has been given to you so we can make something exciting for your first lesson.

To make things a bit more challenging, some of the code has been excluded from the code 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. **You don’t have to include them in your code.**

Remember to indent code where prompted (->). Use four spaces for each indent!

```
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

while guessedNum != randomNumber: # code inside code block repeats until guessed number is correct

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

-> else: # if guessed number greater than random number, print "Guess Lower!"
->-> _____

-> guessedNum =_____  # USER INPUTS another guess
_____ # user has guessed correctly, print "You Won!"
```

Tasks:

1. Fill in the blanks in the above code.
2. Save the program.
3. Run the program!

Extra Tasks

Output Formatting

Sometimes we would like to format our output to make it look attractive. This can be done by using the string.format() method. Try out these examples:

```
x = 2
```

```
y = 5

print("The value of x is {} and y is {}".format(x,y)) # prints "The value of x is {} and y is {}"
```

```
print("I love {0} and {1}".format("tea","toast")) # prints "I love tea and toast"

print("I love {1} and {0}".format("tea","toast")) # prints: "I love toast and tea"
```

Tasks:

1. In the **first example** code block above, print out the value of x and y, switching the order of x and y in str.format().
2. In the **second example** code block above, print out the sentence "I love toast and tea" (change numbering in curly brackets ("{}").

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" # String variable

myInt = (int)(myString) # converted and stored in Int variable
```

Tasks:

1. Store an **int** variable in a **float** (decimal number) variable.
2. Store a **float** value in a **string** variable.

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

Additional Information

Variables in Python

Unlike a lot of languages, Python allows us to declare variables without having to initially assign a variable type. See the example below for declaring a variable and assigning it a value in Python versus doing the same in Java.

Python	**Java**

myNumber = 5	int myNumber = 5;

White Space

In languages such as C++ or Java, curly brackets (“{}“) are generally used to separate blocks of code e.g. within IF statements and FOR loops etc. However, in Python we use white space to separate code blocks. Typically the user will create four spaces, indenting the code, in order to begin the block of code and will then backspace four times when the user wishes to exit the block of code.

Python	**Java**

if ballHozPoz < 800:
  ballHozPoz = ballHozPoz + 1
else:
  ballHozPoz = -50
	if (ballHozPoz < 800) {
	  ballHozPoz = ballHozPoz + 1;
	}
	else {
	  ballHozPoz = -50;
	}

for i in range(5):
  j = i*i
  print(i, j)
	for(int i = 0; i < 5; i++) {
	  int j = i*i;
	  println(i, j);
	}

Careful! Do not add in any additional spaces or tabs to further indent any line in the code block. Python needs each line in the code block to be exactly aligned so it can read the code.