# **Introduction to python using dice**

At some points during the week we will be using python to do some maths. Here is an introduction to cover some of the basic concepts you will need. If you are already familiar with python you could try to do the Dice Simulation and Pig game yourself without bothering with the instructions!

You can use this website <https://trinket.io/home> to test and run python code.

# Dice Simultation

When you throw two dice and add the numbers together what number are you most likely to get? What is the smallest number you can get? What is the biggest? Are all numbers equally likely?

Use python to simulate rolling a dice 100 times and plot your results using pygal.

# Pig Game

The game of Pig is a two player game played with two six-sided dice. The object of the game is to reach 100 points of more. Play is taken in turns. On each person's turn that person has the option of either:

1. **Rolling the dice:** where a roll of two to six is added to their score for that turn and the player is given the same choice again; or a roll of 1 loses the player's total points *for that turn* and their turn finishes with play passing to the nexxt player.

2. **Holding:** the player's score for that round is added to their total and becomes safe from the effects of throwing a 1. The player's turn finishes with play passing to the next player.

Create a program to score for, and simulate dice throws for this game

Extension - What about if two dice are rolled each time and a one on either dice gives zero for that turn, and a double 1 wipes out all previous scoring.

# Dice Simulation Instructions

## Task 1

Go to <https://trinket.io/home> , sign up for an account, and then type the following code into a new python trinket



Press the play button to run the code. You should see the numbers 2 to 198 appear on the right. The \* symbol means multiply.

Try changing the 2\*x to 5\*x

Try changing the 2\*x to 10

Can you generate the sequence 6,12,18,24,…,180?

## Task 2

from random import randint

The above code is a command which goes to a module (a collection of mini programmes) and allows you to use the one called randint.

randint(5,10) would, for example generate a random integer between 5 and 10.

Type in

from random import randint

x=randint(5,10)

print (x)

Press play a few times and see what you get.

x is a “variable” in the code above. You have named a variable x and set it equal to a random integer between 5 and 10 (inclusive).

Change the code slightly to simulate a dice being thrown. Call the variable dice1.

## Task 3

Use this idea to simulate two dice being thrown – dice1 and dice2.

Create a new variable called total and set it equal to the sum of the two dice.

Print the total.

## Task 4

Using everything you have learnt so far, write a program that does Task 3 100 times

## Task 5

You should have this so far:



Add the following two lines (each time you have a : you need to indent the next line, i.e. have at least one space)



Run the program a few times and see what happens.

Now instead of saying “I got a 2!” we are going to count how many times this happens.

On line 2 of your code, add in a new variable and set it equal to 0.



Then replace **print(“I got a 2!”)** with **twos=twos+1**

At the end of your code (not indented), write **print(twos)**. This should tell you how many twos you got.

## Task 6

Add all the code needed to say how many threes you got.

## Task 7

It would take a long time to do this for every number, so instead we can use code like this:



In this code, frequencies is an example of a list. It stores lots of variables at once. You can access each variable using, for example, frequencies [5]. This would give you the 5th variable (well, the 6th actually because the numbering starts at 0!). Run the code above and answer the following questions on the result you get:

1. What do your numbers add up to? Why
2. Why do you have two zeroes at the start of list?
3. What was the most common number you rolled, and how many times did you roll it?

## Task 8

Imports a module to allow you to draw graphs



Draws the graph

Adds a line segment from the coordinate (2,0) to another coordinate, whose y value depends on how many 2’s have been thrown

Creates a new graph and then gives it a title

Run the code above and see what is produced

## Task 9

In a slow way or a quick way, plot one graph showing all the frequencies from 2 to 12.