

# List of Concepts

April 28, 2020

## 0.1 *Variable assignment and algebraic manipulation*

This code assigns the values 5 and 2 to a and b respectively, then prints the value of 5 mod 2.

```
[1]: a = 5
      b = 2
      print(a % b)
```

1

## 0.2 *Single conditional statements*

This code checks the sign of a random integer between -5 and 10.

```
[7]: import random

n = random.randint(-5, 10)
if n < 0: #checks if n is negative
    print("n = ", n, ". n is negative.")
elif n > 0: #checks if n is positive
    print("n = ", n, ". n is positive.")
else: #if n is neither positive nor negative, it must be zero
    print("n is zero.")
```

n = 7 . n is positive.

## 0.3 *Repeated conditional statements*

This code flips a coin until it gets heads, and checks how many tails it gets before this.

```
[42]: import random

tails = 0
heads_achieved = False

while heads_achieved == False: #repeat until a head is achieved
    if random.randint(0, 1) == 1: #where 1 is a head
        print("Heads!")
        heads_achieved = True #stops the loop
    else: #else, it is a tails; repeat loop
```

```

        print("Tails!")
        tails += 1
    print(tails)

```

```

Tails!
Tails!
Tails!
Tails!
Tails!
Tails!
Heads!
6

```

## 0.4 *Functions and lists*

This function creates a list of all even numbers smaller than a given number.

```

[29]: def evens(n):
        '''Creates a list of all even numbers smaller than n'''
        results = [] #creates an empty list in which to put the numbers
        for i in range (n):
            if n > 2 * i: #where 2 * i is the even sequence 2,4,6...
                results.append(2 * i)
        return results

```

```

[30]: evens(22)

```

```

[30]: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

```

## 0.5 *Iteration*

This code checks how many vowels are in a string.

```

[3]: word = "Dublin, Ireland"
    letters = list(word.lower())
    vowels = 0

    for n in range (len(word)): #loops for as many letters as the list contains
        if letters[n] in ['a', 'e', 'i', 'o', 'u']: #if the letter being checked is
            ↪ a vowel
            vowels += 1
    print(vowels)

```

```

5

```

## 0.6 *Recursion*

This function finds the n'th triangular number.

```
[43]: def triangular(n):  
    '''Finds the nth triangular number by recursion.'''  
  
    if n <= 1:  
        return 1  
    else:  
        return n + triangular(n-1)
```

```
[44]: triangular(5)
```

```
[44]: 15
```

## 0.7 Object-oriented programming

This class represents vectors in 3-dimensional space.

```
[64]: class Vector:  
    '''A class for vectors in 3-dimensional space.'''  
  
    def __init__(self, x, y, z):  
        self.x = x  
        self.y = y  
        self.z = z  
  
    def __add__(self, other): #adds two vectors together  
        return Vector(self.x + other.x, self.y + other.y, self.z + other.z)  
  
    def dot_product(self, other): #gives the scalar (dot) product of two vectors  
        return (self.x * other.x + self.y * other.y + self.z * other.z)  
  
    def cross_product(self, other): #gives the vector (cross) product of two  
    ↪ vectors  
        return Vector(self.y * other.z - self.z * other.y, self.z * other.x -  
    ↪ self.x * other.z, self.x * other.y - self.y * other.x)  
  
    def __repr__(self): #represents vectors as a string, rather than an object  
        return str([self.x, self.y, self.z])
```

```
[65]: A = Vector(2, 3, 4)  
      B = Vector(1, 5, 1)
```

```
[66]: A + B
```

```
[66]: [3, 8, 5]
```

```
[67]: A.dot_product(B)
```

[67]: 21

[68]: `A.cross_product(B)`

[68]: [-17, 2, 7]

[70]: *#this should be the inverse of  $A \times B$*   
`B.cross_product(A)`

[70]: [17, -2, -7]