# What will you learn?

* Introduction to the Python programming language
* Introduction to simple logic and python programming constructs
  + Module
  + Class
  + Variable
  + Method
  + Conditional (If/then/Else)
  + Loops (For)
* How to write your first game in Python

# Prerequisites

Install Pycharm - https://www.jetbrains.com/pycharm/download/#section=windows  
Install Python - https://www.python.org/downloads/  
Instal Pygame - https://www.pygame.org/

# Rules of the Snake Game

* The red blocks represent the snake and the green block represents the apple.
* The goal of the game is for the snake to eat as many apples as possible without colliding into itself.
* It starts out easy and becomes more difficult as the length of the snake grows.
* The snake can be moved using the right, left, up and down arrow keys.

# Building our Snake Game in Python

## Import Python Modules

**pygame module:** The pygame module is a cross-platform set of Python **modules** designed for writing video games. It includes computer graphics and sound libraries.  
**os module:** The OS module in Python provides a way of using operating system dependent functionality.  
sys module: The sys module allows you to use stdin() and stdout(), as well as stderr().

1. *Import the pygame Module*

# Create an import statement to import a python module named **pygame.**

Import pygame

*2. Create the Apple Class*

**# The Apple Class defines the Apple in our game.  
# When the Snake eats (touches) the Apple, the Snake will grow.  
# The Apple class will set the starting location of the Apple.****class** Apple:  
 x = 0  
 y = 0  
 step: int = 44  
  
 **def** \_\_init\_\_(self, x, y):  
 self.x = x \* self.step  
 self.y = y \* self.step  
  
 **def** draw(self, surface, image):  
 surface.blit(image, (self.x, self.y))

*3. Create the Snake Class*

**class** Snake:  
 x = [0]  
 y = [0]  
 step = 44  
 direction = 0  
 length = 3  
 windowWidth = 800  
 windowHeight = 600  
 updateCountMax = 2  
 updateCount = 0  
  
 **def** \_\_init\_\_(self, length):  
 self.length = length  
 **for** i **in** range(0, 2000):  
 self.x.append(-100)  
 self.y.append(-100)  
  
 *# initial positions, no collision.* self.x[1] = 1 \* 44  
 self.x[2] = 2 \* 44  
  
 **def** update(self):  
  
 self.updateCount = self.updateCount + 1  
 **if** self.updateCount > self.updateCountMax:  
  
 *# update previous positions* **for** i **in** range(self.length - 1, 0, -1):  
 self.x[i] = self.x[i - 1]  
 self.y[i] = self.y[i - 1]  
  
 *# update position of head of snake* **if** self.direction == 0: *# Move Right* self.x[0] = self.x[0] + self.step  
 **if** self.windowWidth < self.x[0]:  
 self.x[0] = self.windowWidth - self.x[0]  
  
 **if** self.direction == 1: *# Move Left* self.x[0] = self.x[0] - self.step  
 **if** self.x[0] < 0:  
 self.x[0] = self.windowWidth - self.x[0]  
  
 **if** self.direction == 2: *# Move Down* self.y[0] = self.y[0] - self.step  
 **if** self.y[0] < 0:  
 self.y[0] = self.windowHeight + self.y[0]  
  
 **if** self.direction == 3: *# Move Up* self.y[0] = self.y[0] + self.step  
 **if** self.y[0] > self.windowHeight:  
 self.y[0] = self.windowHeight - self.y[0]  
  
 self.updateCount = 0  
  
 **def** moveRight(self):  
 self.direction = 0  
  
 **def** moveLeft(self):  
 self.direction = 1  
  
 **def** moveUp(self):  
 self.direction = 2  
  
 **def** moveDown(self):  
 self.direction = 3  
  
 **def** draw(self, surface, image):  
 **for** i **in** range(0, self.length):  
 surface.blit(image, (self.x[i], self.y[i]))

1. *Identify the Variables in the Snake Class*
2. *Identify the Methods in the Snake Class*
3. *Identify the If/then/else*
4. *Identify the For Loop*
5. *Identify the WhileLoop*