



Teaching Recursion to Junior-High School Students by Using Fractals: A Complete Lesson Plan in Python

Vasileios Drakopoulos, Panagiotis-Vlasios Sioulas

Department of Computer Science and Biomedical Informatics, Faculty of Science, University of Thessaly, Lamia, Greece

Email address:

vdrakop@uth.gr (V. Drakopoulos), psioulas@uth.gr (P.-V. Sioulas)

To cite this article:

Vasileios Drakopoulos, Panagiotis-Vlasios Sioulas. Teaching Recursion to Junior-High School Students by Using Fractals: A Complete Lesson Plan in Python. *American Journal of Education and Information Technology*. Vol. 4, No. 2, 2020, pp. 50-55.

doi: 10.11648/j.ajeit.20200402.12

Received: April 8, 2020; **Accepted:** May 8, 2020; **Published:** July 17, 2020

Abstract: Recursion and functional programming are two programming techniques that go beyond the framework requirements but they are fundamental to learning to program. Recursion is an important idea in computer science and has traditionally been a difficult concept for students to understand, both as a control structure and as an analytic tool. Comprehension of the way programmes are developed bears a number of obstacles especially for 3rd Grade Junior High School students who need to get accustomed to recursion. A one-teaching-hour lesson plan intended for pupils of the 3rd Grade of Junior High School about teaching recursion through python in combination with its turtle library is proposed. The teaching proposal of the specific method utilises a special category of sets which are called fractals. Since the students will be familiarised with a difficult programming technique without, however, being taught mathematical concepts that are difficult to understand, it is expected to have a more positive outlook towards key concepts and in turn to programming. With the introduction of this approach, students acquired understanding of the concept of coding recursion and applied it in the higher-level programming. In addition, our teaching approach made students enthusiastic, motivated and engaged with the learning of usually difficult subjects.

Keywords: Recursion, Geometry, Computer Graphics, Fractal, Python

1. Introduction

Recursion is one of the most popular approaches in developing programmes. It is based on the capacity of every contemporary programming language during which a process or a function calls itself [1].

One of the simplest processes of calculating the number of a classic sum is to count its constituents. In case the constituents of the sum are classified in a series of subsets their calculation is not feasible. The sole exception is that of subsets whose constituents derive having some kind of relation to the constituents of those immediately preceding them. This relationship is called recursion [2].

Recursion as a theory is cited in references dating back to the 1930s and we may find it in works written by Gödel [3], Church [4], Turing [5], Kleene [6] and Emil Post [7]. According to international bibliography, recursion is the process of classification of subsets each of which is created due to its relation to the subsets preceding it.

Effective strategies for introducing the topic include

using different contexts such as recurrence relations, programming examples, fractal images and a description of how recursive methods are processed using a call stack. A number of results and advices coming from our observations and didactical experience gathered when teaching about recursion in different contexts and on various education level (K-12 and tertiary) are discussed in [8]. The article [9] surveys the computing education research literature and presents findings on challenges students encounter in learning recursion, mental models students develop as they learn recursion, and best practices in introducing recursion.

The present study consists of two parts. In the first part, we make a brief description and presentation of fractals while in the second part the proposal for teaching recursion based on fractals in secondary education and more specifically in the third grade of Junior high School is analysed.

2. Fractal

2.1. What a Fractal Is

Many natural and artificial phenomena have the very fundamental characteristic of invariance under different scales, have infinite details at every point, are self-similar across different scales and can be described by a procedure that specifies a repeated operation for producing the details. The term ‘fractal’ was introduced during the 1970s by Benoit Mandelbrot, a French of Polish descent mathematician. *Fractal* comes from the Latin adjective *fractus*, which has the same root as *fraction* and *fragment* and means “irregular or fragmented”; it is related to *frangere*, which means “to break”.

What distinguishes fractals from classical geometric figures is their *dimension*. The notion of dimension is very familiar, but surprisingly subtle. Intuitively, we know that a line or curve is one-dimensional, a plane or surface is two-dimensional and space is 3-dimensional.

A fractal was defined by Mandelbrot as a subset whose Hausdorff - Besicovitch dimension is greater than its topological or intuitive dimension [10], where Hausdorff - Besicovitch dimension is a measure of *roughness*. There are several ways to define a *fractal dimension* like the above-mentioned one, but they are not always equivalent. Since the dimension could be used to quantify an aspect of the ‘form’, the term is featured in Greek language as ‘morphoklasma’ enriching the aforementioned definition with some features.

According to Falconer [11], a sum of points bear new details in every scale of its size. He added that a fractal is characterized by the feature of self-similarity which means that one of its subsets is similar to another regardless of any change in scale. This particular feature is significant to fractals [12]. Last but not least, a fractal could be formed through a recursive process – applying the same approach – of every single step; see for example Figure 1.

The emergence of fractals brought secrets to light concerning the geometrical complexity of nature and contributed to forming new terminology through which we are able to describe among other things clouds, trees and sponges

in a similar way we would describe the branches of our blood vessels and lungs [13].

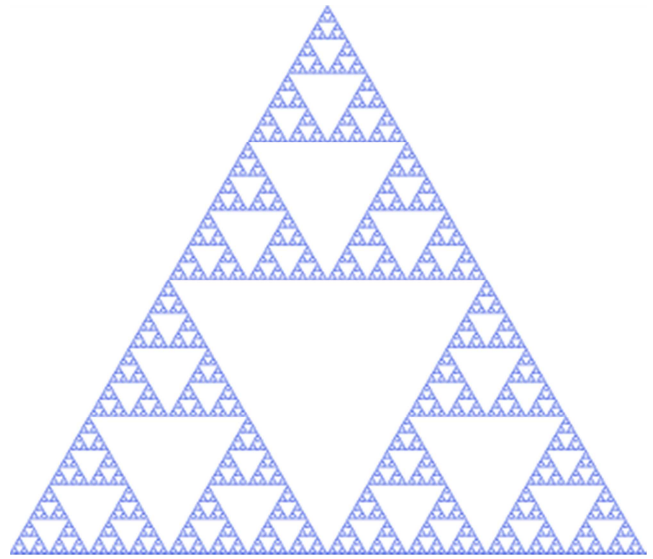


Figure 1. The Sierpiński triangle—a confined recursion of triangles that form a fractal.

2.2. Self-Similarity

An object is regarded as self-similar provided its constituents are similar to the whole. The repetition of formations takes place gradually in smaller scale and it is likely for individual parts – if magnified – to look similar to the whole object while at the same time remains unchanged under scale changes [14]. The aforementioned features are found in snowflakes, barks of trees, coastlines, deserts and elsewhere; see Figure 2.

However, there are neither identical trees nor clouds as there are no identical rocks, coasts, deserts and galaxies whose form and dimensions coincide. The natural world is multiplied tenaciously, patiently and by having a memory of the archetype like a fern leaf that copies the same shape in every single of its parts, like the reflection of the shape itself in the mirror, endlessly [15].



Figure 2. Fractal patterns in nature and art; a fern, a tree and a snowflake.

2.3. Pedagogical Implications and Utilisation

Having acquired a deal of knowledge throughout primary school, students have formed a positive stance towards the use of computers. However, in Junior High School they

merely come up against programming concepts (sequential structure, etc.) which they are asked not only to comprehend but also be capable of realizing in a programming environment. The outcome of such demands leads them to abruptly realize they cannot cope with them since there

appears to be a shift from learning some applications to actually facing a challenging aspect of Informatics which is Programming.

What remains of great value, however, is the magic world of colours and graphics. Most students in Junior High Schools are keen on using Paintbrush compared to Word Processing, Spreadsheets or even Databases. Thus, learners tend to opt for creativity. In our days, it has been acknowledged that image plays an imperative role as it is considered to be the most useful means of description in complicated abstract relations [15].

Due to their impressive shapes, fractals make quite an impression on learners and draw their attention instantly. This visual expression awakens them and fulfills their sensor quests since it is a fact that beautiful pictures impress learners. The process of teaching basic programming concepts via the use of fractals could be considered an unprecedented object that gives learners the opportunity to creatively interact with programming and geometry simultaneously. Moreover, they are provided with the chance to create shapes and images that they have never thought they could create via programming. It should be stressed that they have the means to investigate classic aspects of programming through the implementation of an interesting approach.

These multilayered creative potentials result in educators having a wide range of choices in order to enrich their teaching along with their material. The geometry of fractals is a new language used to describe, provide standards as well as analyse the complicated forms found in nature. Moreover, it is the most suitable environment for improving mathematics education since it is applied in an open learning environment creating activities through which learners develop concepts through deductive reasoning into productive by experimenting. The most characteristic application is that of the Photodentro Learning Object Repository, or LOP for short. It constitutes a core part of the Greek Ministry of Education and Religious Affairs' digital infrastructure for education content for schools. 'Introducing Fractals'¹ is a research activity aimed at familiarising students with the concept of Fractal. Students discover the properties of fractals and produce them using a repetitive process.

2.4. Binary Tree

A classic example of a fractal shape is the binary tree (Figure 3). The specific shape starts from a vertical straight part that represents the main trunk of a tree. From the top of the trunk start two straight parts (branches) bearing the same length so as to form the same angle with the straight part. The process is repeated in the edges of each branch so as to maintain the ratio for the branches and for the angles to be equal.

Geometrically, fractals have forms, or features, that repeat at different sizes over ranges of scales. This quality is

equivalent to the concept of recursion. Therefore, there appears to be a possibility to teach in different ways without using complicated mathematic forms and avoid examples such as the Fibonacci sequence as well as that of finding the greatest common divisor. In the next section a complete lesson plan is presented for teaching recursion with the use of fractal sets.

3. The proposed Lesson Plan

The present section presents a complete lesson plan intended for one teaching hour during which an introduction to recursion is performed. At this point, it should be highlighted that we take for granted that students have been taught sub-programmes (processes) in python.

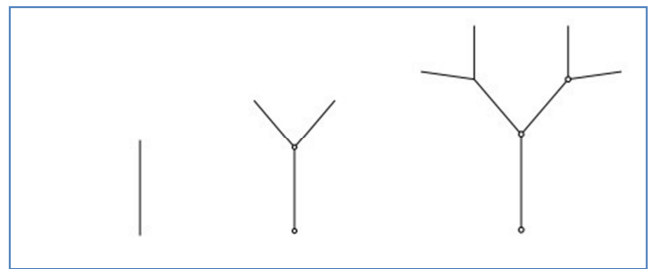


Figure 3. A binary tree.

3.1. General Data

Course: Informatics for 3rd Grade of Junior High School
Title

1. 'Teaching the concept of recursion through geometrical shapes with the use of python'.

Part A General Data-Goals-Means

1. Grade: 3rd, Junior High School.
2. Teaching hour (s): 1.

Goal

1. The student creates and explores the geometrical structures by making use of recursion.

Teaching Approach

1. Enriched lecture, collaborative learning, exploratory instruction.

Goals

The student should

1. Be able to design small shapes using recursion.
2. Comprehend the function of basic commands of recursion.
3. Be able to set parameters for parts of a code.
4. Be able to change parameters in a function or process.
5. Identify that shapes of the Euclidean Geometry also have self-similarity [16].

Teaching Means

1. Computer, projector, Whiteboard, pycharm software or similar environment using python.

References

1. Third Grade of Junior High School Informatics textbook [17].

¹ <http://photodentro.edu.gr/lor/r/8521/8003?locale=en>.

3.2. Teaching

Warm-up - Connection with previous session (5')

1. Learners are asked questions concerning the function of python processes.

2. Learners are divided in teams of two or three members.

Presentation (35').

1. Presentation of the concept of recursion by using Figure 4 and Figure 5.
2. Students are provided with part of a code shown in Figure 6 and are asked to type in pycharm environment and 'run' it.
1. They are given 5' to write down how they conceptualize running the algorithm they have typed.
2. They are asked to show which is/are the recursive command (s).
3. They are presented with the notion of fractal sets.
4. Finally, the worksheet is distributed.

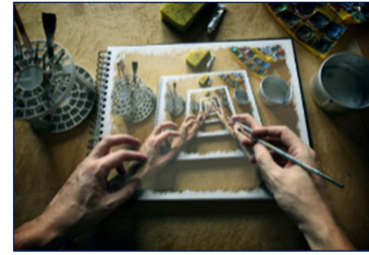


Figure 4. Droste effect.



Figure 5. Broccoli.

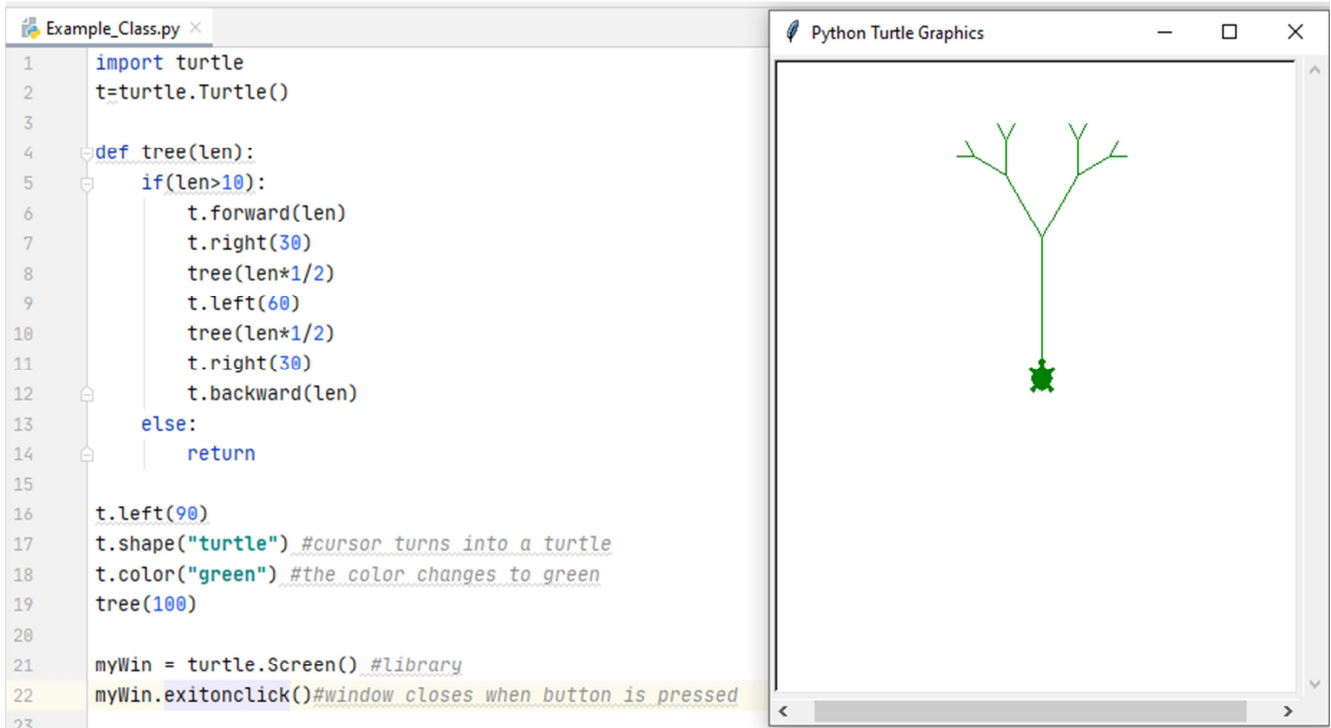


Figure 6. Recursion Example in Python.

Summary (1')

Evaluation (5')

1. Why is recursion used?
2. If you change the value of len variable, what do you observe happening?

3.3. Worksheet

Exercise 1: Type the code provided so that you can create the tree in Figure 7.

Exercise 2: Write down the figure provided above so as to

comprehend its function.

Exercise 3: Create your own variations by changing: a. limit 10 in the execution terminating condition, b. the fraction that defines which part will be used in every call, c. the angle its branches form. With regards to the angle you should pay attention to the value you assign and you will find it on your own.

Homework

Task 1: Type the code given so as to create the tree in Figure 8.


```
import turtle
t = turtle.Turtle()
def tree(len):
    if(len>10):
        t.forward(len)
        t.right(30)
        tree(len * 2 / 3)
        t.left(60)
        tree(len * 2 / 3)
        t.right(30)
        t.backward(len)
    else:
        return
    t.left(90)
    tree(100)
myWin = turtle.Screen()
myWin.exitonclick()
```

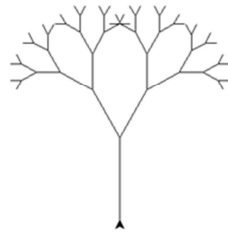


Figure 7. First Exercise's code execution.

```
import turtle
t = turtle.Turtle()
t.left(90)
def tree(len):
    if(len>10):
        t.forward(len)
        t.right(40)
        tree(len * 2 / 3)
        t.left(60)
        tree(len*2/3)
        t.right(20)
        t.backward(len)
    else:
        return
    tree(100)
myWin = turtle.Screen()
myWin.exitonclick()
```

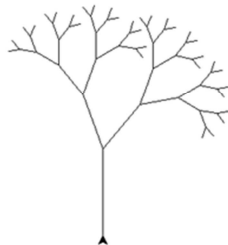


Figure 8. Result of Task 1.

Task 2: Type the code given so as to create the tree of Figure 9.

```
import turtle
t = turtle.Turtle()
t.left(90)
def tree(len):
    if(len>10):
        t.forward(len)
        t.right(45)
        tree(len * 2 / 3)
        t.left(60)
        tree(len / 2)
        t.right(15)
        t.backward(len)
    else:
        return
    tree(100)
myWin = turtle.Screen()
myWin.exitonclick()
```



Figure 9. Result of task2.

Task 3: Create the appropriate parameters in the heading of the tree process so that it will accept the data you want to be altered in order not to be necessary to change its syntax every time.

4. Conclusions

A complete teaching proposal for recursion was presented in the current study by overlooking typical mathematics problems such as the Fibonacci sequence, factorial or finding the greatest common divisor and concentrating on using special geometric shapes that are named fractals. Through the implementation of the specific proposal, the students will visually comprehend how the recursive methodology works which is considered to be one of the most demanding concepts in programming and preoccupies not only learners but also computer science students. The specific teaching process was implemented in python which is considered suitable for familiarising learners with a programming language, but see also [18].

Recursion is a common phenomenon in nature and a basic building block of computation which, however, is rarely taught in schools. Enriching teaching with the use of graphics opens up new potential concerning teaching various lessons since it is scientifically acknowledged that the use of pictures helps in keeping the students alert and helps towards making the lesson more comprehensible for them.

Acknowledgements

An earlier version, written in Greek, of the proposed method in Scratch was published in the 10th Conference on Informatics in Education 2018 which took place in the University of Macedonia in Greece [19].

References

- [1] Athena Vakali, I. Giannopoulos, N. Ioannidis, C. Koiliias, K. Malamas, Y. Manolopoulos and P. Politis, Application development in a programming environment, Athens: Institute of Education Policy, 1999. (in Greek).
- [2] D. Georgiou, E. Antoniou and A. Chatzimichailidis, Discrete Mathematical Structures in Computer Science, Athens: Hellenic Academic Libraries Link, 2015. (in Greek).
- [3] K. Gödel, "Die vollständigkeit der axiome des logischen Funktionenkalküls," Monatshefte für Mathematik, 37 (1), 1930, pp. 349-360.
- [4] A. Church, "A note on the Entscheidungsproblem," The journal of symbolic logic, 1 (1), 1936, pp. 40-41.
- [5] A. M. Turing, "Computability and λ -definability", The Journal of Symbolic Logic, 2 (4), 1937, pp. 153-163.
- [6] S. C. Kleene, "General recursive functions of natural numbers", Mathematische Annalen, 112 (1), 1936, pp. 727-742.
- [7] E. L. Post, "Recursively enumerable sets of positive integers and their decision problems", Bulletin of the American Mathematical Society, 50 (5), 1944, pp. 284-316.

- [8] M. M. Syslo and A. B. Kwiatkowska (2014) Introducing Students to Recursion: A Multi-facet and Multi-tool Approach. In: Gülbahar Y., Karataş E. (eds) Informatics in Schools. Teaching and Learning Perspectives. ISSEP 2014. Lecture Notes in Computer Science, vol 8730. Springer, Cham.
- [9] Renée McCauley, Scott Grissom, Sue Fitzgerald and Laurie Murphy (2015). Teaching and learning recursive programming: a review of the research literature. *Computer Science Education*, 25: 1, 37-66, DOI: 10.1080/08993408.2015.1033205.
- [10] B. B. Mandelbrot, "The fractal geometry of trees and other natural phenomena," in *Geometrical probability and biological structures: Buffon's 200th anniversary*. Springer, 1978, pp. 235-249.
- [11] K. Falconer, "Fractal geometry: Mathematical foundations and applications", 3rd ed., New Jersey: John Wiley & Sons, 2014.
- [12] M. F. Barnsley, "Fractals Everywhere", 3rd ed., Dover Publications, Inc., New York, 2012.
- [13] V. Drakopoulos, "Scientific and artistic creation as assistants in the educational process", *Geometry: From science to application*, 2012, p. 12. (in Greek).
- [14] C. Song, S. Havlin and H. A. Makse, "Self-similarity of complex networks", 433 (7024), 2005, p. 392.
- [15] Leoni Evaggelatou-Dalla and V. Drakopoulos, "The new dimension of educational mathematical thinking", in 14th Panhellenic Conference on Mathematical Education, Hellenic Mathematical Society, 1997, pp. 235-242. (in Greek).
- [16] S. I. Oikonomidis and G. T. Kalkanis, "Repetitive Procedures in Natural Science, in Mathematics and in Informatics Education". in *Didactics of Physical Science and New Technologies in Education*. 2007. (in Greek).
- [17] A. Aggelis, N. Alexandris, P. Georgiadis, K. Gyrtis, A. Kostakos, A. Raptis, and L. Stergiopoulou-Kalantzi, *Third Grade of Junior High School Informatics*, Athens: School Book Publishing Organization (OEDB), 1997. (in Greek).
- [18] B. S. Elenbogen and M. R. O'Kennon, "Teaching recursion using fractals in Prolog", *ACM SIGCSE Bulletin*, 20 (1), 1988, pp. 263-266.
- [19] V. Drakopoulos and P.-V. Sioulas, "Teaching Recursion to Junior-High School Students with the use of fractals: a complete lesson plan" in 10th Conference on Informatics in Education. Thessaloniki, 2018. (in Greek).