

Forum:[Hack Forums](#)

Thread: [Recursion & complexity](#)

The Broker - March 16, 2020, 1:09 pm

Where are the best online sources to learn these topics?

Will +3 good suggestions

Sogomn - March 17, 2020, 7:57 pm

"Introduction to Algorithms" by Cormen et al. covers time complexity pretty well, including recursive algorithms.

There is an online version [here](#). Chapter 3 is probably the most relevant one. I'd suggest to take a look at the different (recursive) sorting algorithms aswell.

Juice. - March 18, 2020, 10:14 am

We haven't quite gotten to recursion my class yet, but I know this site has good practice exercises ranging from basic to more advanced for when you do get some understanding of the topic.

<https://codingbat.com/java/Recursion-1>

SIM-3107656 - March 21, 2020, 10:51 am

I wouldnt learn recursion on Java. I would on C. Since it is not really object-oriented. It is procedural. Best way to learn how to create / use recursion is to try it myself. I can learn you.

MavenGradle - March 22, 2020, 5:46 pm

(Mar 21, 2020 10:51 AM)TJackZone Wrote: [CITE](#) I wouldnt learn recursion on Java. I would on C. Since it is not really object-oriented. It is procedural. Best way to learn how to create / use recursion is to try it myself. I can learn you.

This is the Java subforum, not the C/C++ subforum. I don't see how this is helpful in any way to the OP as it is not a requirement to learn another programming language in order to understand a basic concept such as recursiveness.

Recursive functions are possible in a myriad of programming languages including for example JavaScript, which is arguably even easier to understand and learn. Generally speaking it's not so much the language that's presenting the challenge but rather the logical concept of recursion.

In essence, it always boils down to a certain function calling itself (in order to return a specific value, although this is not a requirement for a function to be recursive). They are frequently used in certain search algorithms, but can generally be implemented for even basic (mathematical) concepts or calculations.

This image represents the basic concept behind recursive functions:

For example, say we have the following method in Java, which is simply a function that will subtract 1 from a number until said number reaches a value of zero::

```
Code private int subtractUntilZero(int x) {  
    if (x < 0) throw new IllegalArgumentException("Argument cannot be smaller than zero!");
```

```
while (x > 0) {  
    x--;  
}  
return x;  
}
```

You could make this into a recursive function by calling the function from within the function itself:

```
Code private int subtractUntilZero(int x) {  
    if (x < 0) throw new IllegalArgumentException("Argument cannot be smaller than zero!");  
    if (x == 0) return x;  
    return subtractUntilZero(x-1);  
}
```

Clearly, the function will call itself in the return statement, and while it's calling itself, it's also subtracting 1 from the initial parameter, until it reaches zero. Granted, there's no real world use for these functions but it's rather useful for demonstrative purposes and might provide the basic insight that is required to understand recursion in computer programming.

Since the concept is so generally applicable, the OP can now take this knowledge and learn C or any other programming language and apply the same concept of recursion. Obviously, the point of this post is that it is not the language that's the problem, it's the understanding of the general theory/concept.