El futuro digital es de todos MinTIC

Week 5 Class 1

Student Worksheet

Main Course

UNIVERSIDAD SERGIO ARBOLEDA

MisiónTIC2022

I. Before you read

A. Keywords

Look at the Word Cloud of the most frequent words in the text you are going to read.

1. **Which words are similar in Spanish?** Write down the unknown words and look them up in a dictionary.

Unknown words:

B. Predicting

2. After checking the vocabulary, write a sentence predicting the topic that you are going to read in this text.

3. Vocabulary. Read the following words, check them in the text, and taking into account their context, match them with the correct meaning.

1. Chipping away		**A.** a series or group of symbols, words, or bits treated as a unit.
2. Break down		**B.** to take something or someone to a specific destination
3. Loop		**C.** to smash, split, or divide into parts violently
4. Drops off		**D.** to cut or break a bit or fragment
5. Grok		**E.** to understand intuitively.

II. While you read

A. Scanning

1. Scan the text and answer the following questions.

a. What´s an iterative algorithm?

_____

b. How can you solve a case that is not simple?

_____

B. Skimming

2. Skim the text and match the sentence in the correct order.

1. What´s the main objective of the text.		**A.** by mastering concepts such as recursive functions and recursive data structures
2.What will the reader learn?		**B.** to provide the conceptual tools necessary to approach problems from this recursive point of view
3. How can we learn?		**C.** how to work with recursion in Python programs

Thinking Recursively in Python

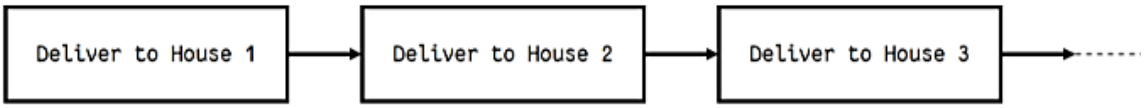
Problems (in life and also in computer science) can often seem big and scary. But if we keep **chipping away** at them, more often than not we can **break them down** into smaller chunks trivial enough to solve. This is the essence of thinking recursively, and my objective in this article is to provide you, my dear reader, with the conceptual tools necessary to approach problems from this recursive point of view.

Together, we'll learn how to work with recursion in our Python programs by mastering concepts such as recursive functions and recursive data structures. We'll also talk about maintaining state during recursion and avoiding recomputation by caching results. This is going to be a lot of fun. Onwards and upwards!

Dear Pythonic Santa Claus…

I realize that as fellow Pythonistas we are all consenting adults here, but children seem to **grok** the beauty of recursion better. So let's not be adults here for a moment and talk about how we can use recursion to help Santa Claus.

Have you ever wondered how Christmas presents are delivered? I sure have, and I believe Santa Claus has a list of houses he **loops** through. He goes to a house, **drops off** the presents, eats the cookies and milk, and moves on to the next house on the list. Since this algorithm for delivering presents is based on an explicit loop construction, it is called an iterative algorithm.



Iterative Present Delivery

The algorithm for iterative present delivery implemented in Python:

```
houses = ["Eric's house", "Kenny's house", "Kyle's house", "Stan's house"]


def deliver_presents_iteratively():
    for house in houses:
```

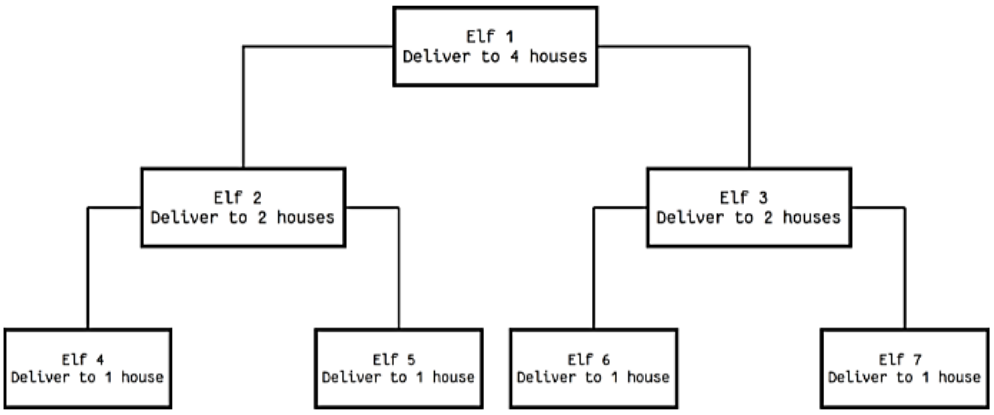
```
        print("Delivering presents to", house)

>>>
```

```
>>> deliver_presents_iteratively()
Delivering presents to Eric's house
Delivering presents to Kenny's house
Delivering presents to Kyle's house
Delivering presents to Stan's house
```

But I feel sorry for Santa. At his age, he shouldn't have to deliver all the presents by himself. I propose an algorithm with which he can divide the work of delivering presents among his elves:

1. Appoint an elf and give all the work to him
2. Assign titles and responsibilities to the elves based on the number of houses for which they are responsible:
   - `> 1` He is a manager and can appoint two elves and divide his work among them
   - `= 1` He is a worker and has to deliver the presents to the house assigned to him



Recursive Present Delivery

This is the typical structure of a recursive algorithm. If the current problem represents a simple case, solve it. If not, divide it into subproblems and apply the same strategy to them.

The algorithm for recursive present delivery implemented in Python:

```
houses = ["Eric's house", "Kenny's house", "Kyle's house", "Stan's house"]

# Each function call represents an elf doing his work
def deliver_presents_recursively(houses):
    # Worker elf doing his work
    if len(houses) == 1:
        house = houses[0]
        print("Delivering presents to", house)

    # Manager elf doing his work
    else:
        mid = len(houses) // 2
        first_half = houses[:mid]
        second_half = houses[mid:]

        # Divides his work among two elves
        deliver_presents_recursively(first_half)
        deliver_presents_recursively(second_half)
```

```
>>>
```

```
>>> deliver_presents_recursively(houses)
Delivering presents to Eric's house
Delivering presents to Kenny's house
Delivering presents to Kyle's house
Delivering presents to Stan's house
```

Retrieven from: https://realpython.com/python-thinking-recursively/

III. After you read

Vocabulary

1. The Word Cloud below shows the most frequent words in the text. Find the words related to **Santa Claus** and circle them.

2. Drawing Conclusions. Why does the author refer to **Santa Claus** in a Python text?

The author says:

Understanding my process

1. Think about your reading comprehension of this article of **Python,** and identify your level of understanding.

a. Good
b. Average
c. Below average

2. What's the most difficult part of reading to me?

a. Vocabulary
b. Grammar
c. Topic

2. Read the following statements and mark Yes or No taking into account your learning process:

Statements	Yes	No
1. I can use strategies to improve my understanding of the text		
2 I can scan a text to find specific information		
3. I can use skimming to get the main ideas		
4. I can use the dictionary correctly to understand the meaning and parts of speech		

Answer Key

I. Before you read

3. Vocabulary. Read the following words and match them with the correct meaning.

1. Chipping away	D	**A.** a series or group of symbols, words, or bits treated as a unit.
2. Break down	C	**B.** to take something or someone to a specific destination
3. Loop	A	**C.** to smash, split, or divide into parts violently:
4. Drops off	B	**D.** to cut or break a bit or fragment
5. Grok	E	**E.** to understand intuitively.

II. While you read

1. Scan the text and answer the following questions.

a. What´s an iterative algorithm?

Since this algorithm for delivering presents is based on **an explicit loop construction,** it is called an iterative algorithm

b. How can you solve case that is not simple?

If the current problem represents a simple case, solve it. If not, **divide it into subproblems and apply the same strategy to them**.

A. Skimming

1. Skim the text and match the sentence in the correct order.

1.Dictionaries are sometimes found in other languages	C	a. dictionaries are indexed by *keys*
2.Unlike sequences, which are indexed by a range of numbers,	A	b. since lists can be modified in place using index assignments
3. Tuples can be used as keys	D	c. as “associative memories”
4. You can’t use lists as keys,	B	d. if they contain only strings, numbers, or tuples

B. Skimming

2. Skim the text and match the sentence in the correct order.

1. What´s the main objective of the text.	B	**A.** by mastering concepts such as recursive functions and recursive data structures
2.What will the reader learn?	C	**B.** to provide the conceptual tools necessary to approach problems from this recursive point of view
3. How can we learn?	A	**C.** how to work with recursion in Python programs

III. After you read

Vocabulary

1. The Word Cloud below shows the most frequent words in the text. Find the words related to Santa Claus and circle them.

2. Drawing Conclusions. Why does the author refer to **Santa Claus** in a Python text?

The author says: “So let’s not be adults here for a moment and talk about how we can use recursion to help Santa Claus”.

The author refers to Santa Claus to give a vivid example of the recursion in Python.