

# PYTHON CONDITIONS AND CLAUSES - IF ELIF ELSE

CSC 109 - Introduction to programming in Python - Summer 2023

Marcus Birkenkrahe

June 6, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Code blocks in other languages</b>	<b>4</b>
<b>3</b>	<b>If statements</b>	<b>5</b>
<b>4</b>	<b>Else statements</b>	<b>5</b>
<b>5</b>	<b>Comparison with C/C++ and R</b>	<b>6</b>
<b>6</b>	<b>Elif statements</b>	<b>8</b>
<b>7</b>	<b>If, elif and else</b>	<b>10</b>
<b>8</b>	<b>Summary</b>	<b>15</b>
<b>9</b>	<b>Glossary</b>	<b>16</b>
<b>10</b>	<b>References</b>	<b>16</b>

## 1 Introduction

- Flow control statements consist of
  1. a *condition* (expression)
  2. a *clause* (block of code)

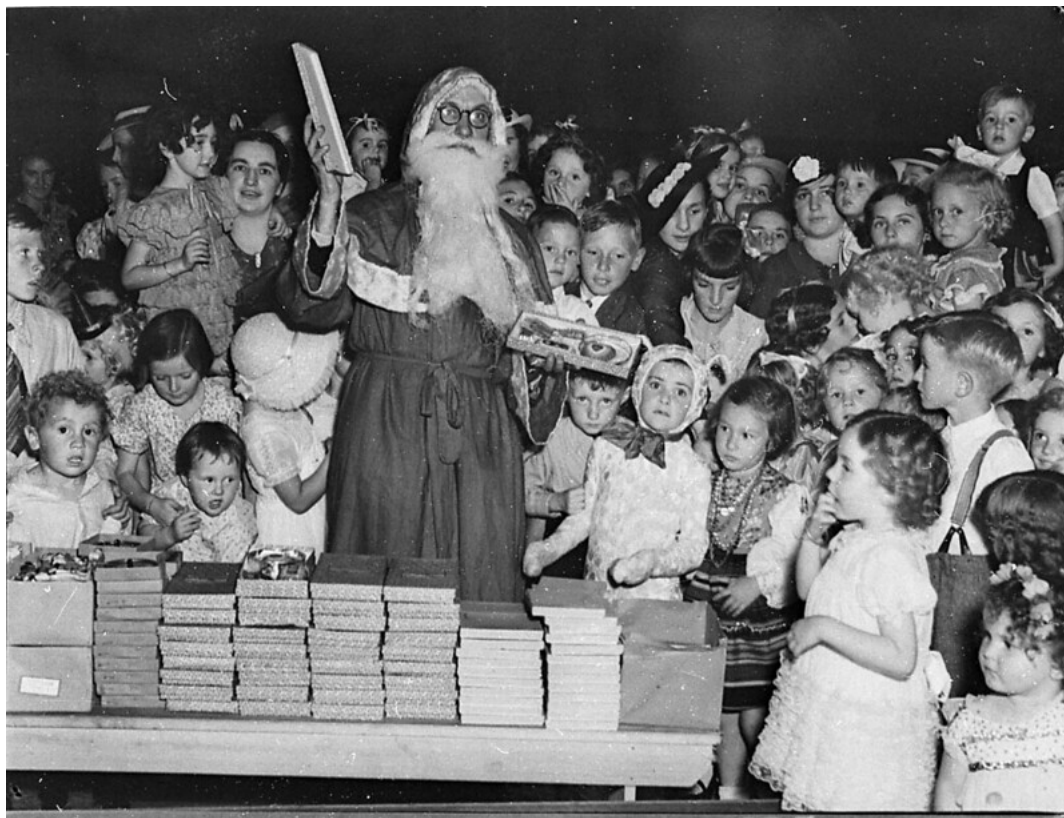
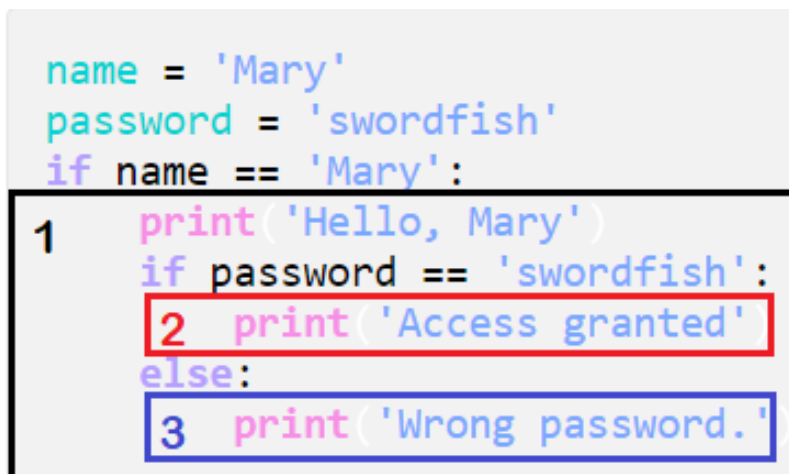


Figure 1: Radio station Children's Christomas Party 1936 (Flickr.com)

- A *condition* is an *expression* that evaluates to a Boolean value and determines if the program enters the *clause* or not.
- Lines of Python code can be grouped into [indented] blocks:
  1. Blocks begin when the indentation increases.
  2. Blocks can contain other blocks.
  3. Blocks end when the indentation decreases to zero or to the indentation of a containing block.
- Example: what are the blocks in this small program?

```
name = 'Mary'
password = 'swordfish'
if name == 'Mary':
    print('Hello, Mary')
    if password == 'swordfish':
        print('Access granted')
    else:
        print('Wrong password.')
```

- In the image, the blocks are highlighted: each colon : marks the beginning of a new block.



```
name = 'Mary'
password = 'swordfish'
if name == 'Mary':
1  print('Hello, Mary')
   if password == 'swordfish':
2  print('Access granted')
   else:
3  print('Wrong password.')
```

Figure 2: Python code blocks with indentation

- View the execution of this program at [autbor.com/blocks/](http://autbor.com/blocks/):
  1. Step through the program with **Next**
  2. The **Frames/Objects** area shows variables and their values
  3. The **Print output** area shows the program output
- Click on **Edit this code** and change **password** to `''` (empty string).
- View the execution again and notice the differences.
- An alternative to [pythontutor.com](http://pythontutor.com) is the built-in debugger **pdb**: you start it for a file `file.py` from the command line with:

```
python -m pdb file.py
```

You can now step through the program with `n` and print variables with `print`.

## 2 Code blocks in other languages

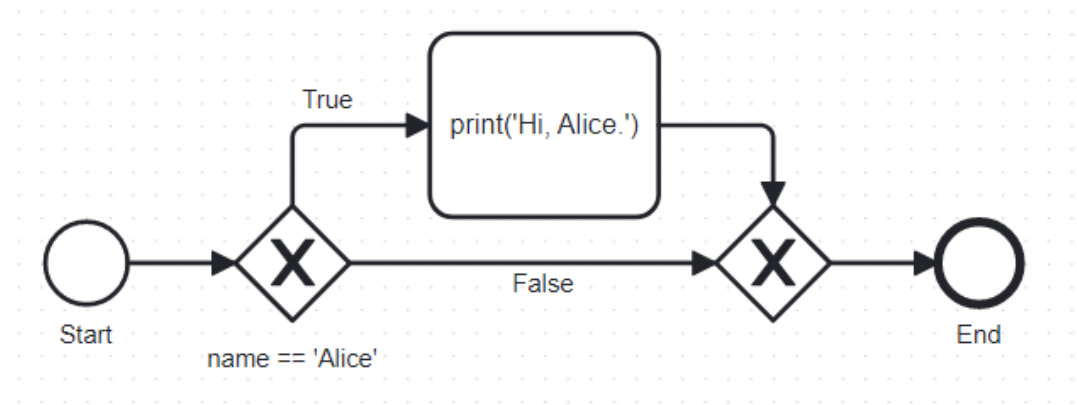
- It is good programming practice in all languages to use indentation to mark code blocks, simply to make code more readable.
- All modern IDEs and editors offer indentation and syntax highlighting, including parenthesis, keywords, variables etc.
- Very few other languages but Python use indentation as a significant feature (Yunits, 2019). They include: Python, Haskell, COBOL, Markdown [Colab, GitHub], Org [Emacs], yaml ['Yet Another Markup Language'].
- Org-mode example: `C-M-\` to indent all lines, or `<TAB>` for one line:

```
name = input('What is your name? ')
if name:
    print(f'Hello, {name}.')
else:
    print('Hello, stranger.')
```

### 3 If statements

- Let's look at a simple example program with and its process diagram:

```
name = 'Alice' # declare a variable 'name'
if name == 'Alice':
    print('Hi, Alice.')
```



- Note the fact that (unlike in a flowchart) the gateway, once opened, needs to be closed again to mark the end of the logical operation. One practical effect is that it is easier to remove decisions from processes because their beginning and end is clear.

### 4 Else statements

- An **if** statement can be followed by an **else** statement whose clause is executed only if the **if** statement's condition is **False**.
- The **else** statement does not have a condition. Let's offer a greeting to a person that is not Alice:

```
if name == 'Alice':
    print('Hi, Alice.')
else:
    print('Hello, stranger.')
```

Hello, stranger.

- **Exercise:** Go to bpmn.io to create a BPMN diagram for this process!
- Solution diagram:

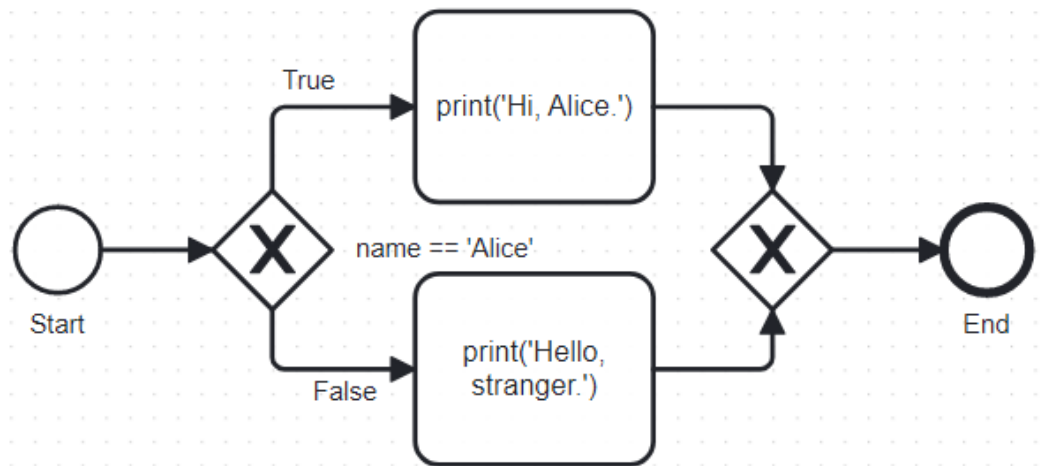


Figure 3: BPMN diagram for an if...else program

- Contrast this result with a non-standard flow chart diagram:

## 5 Comparison with C/C++ and R

- In C/C++, a compiled language (that requires a compiler like `gcc` to generate a machine code file that is then executed), the last program would look like this:

```

#include <stdio.h>
#include <string.h>

int main() {
    char name[] = "Alice";
    if (strcmp(name,"Alice") == 0) {
        puts("Hi, Alice.");
    } else {
        puts("Hello, stranger.");
    }
    return 0;
}

```

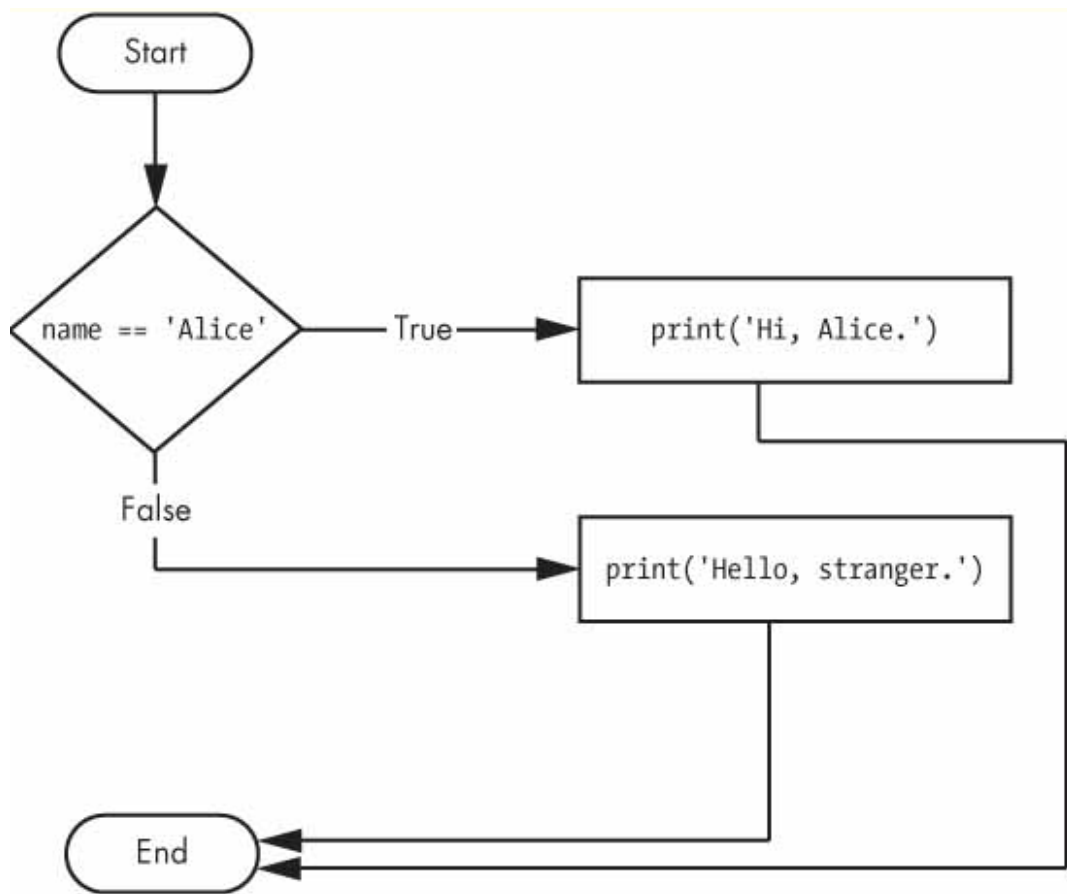


Figure 4: Flowchart for an else statement (Sweigart, 2019)

```
}
```

```
Hi, Alice.
```

- In R, another interpreted language (with a console that executes the code block), it would look like this:

```
name <- "Alice"
if (name == "Alice") {
  paste("Hi, Alice.")
} else {
  paste("Hello, stranger.")
}
```

- Neither of these languages require indentation. They could look like this and would still run - they're not very readable this way:

```
#include <stdio.h>
#include <string.h>
int main() { char name[] = "Alice"; if (strcmp(name,"Alice") == 0) {
    puts("Hi, Alice."); } else { puts("Hello, stranger."); } return 0; }

name <- "Alice"
if (name == "Alice") { paste("Hi, Alice.") } else {
    paste("Hello, stranger.") }
```

## 6 Elif statements

- If you want one of many possible clauses to execute, you can use an "else if" statement called **elif** in Python.
- An **elif** statement consists of:
  1. The **elif** keyword
  2. A condition that evaluates to **True** or **False**
  3. A colon :
  4. An indented clause on the next line
- **Exercise:**



1. Create a new section in your Python Colab notebook and title it **Elif example**.
2. Alter the `if... else "Alice"` program by an `elif` block that prints "You're not Alice, kiddo" if the variable `age` is less than 12.
3. To test the flow, copy your code into `pythontutor.com` and run it with the following values:
  - (a) `name = 'Carol' age = 10`
  - (b) `name = 'Carol' 'age = 28'`
4. Here is the BPMN diagram for this program:

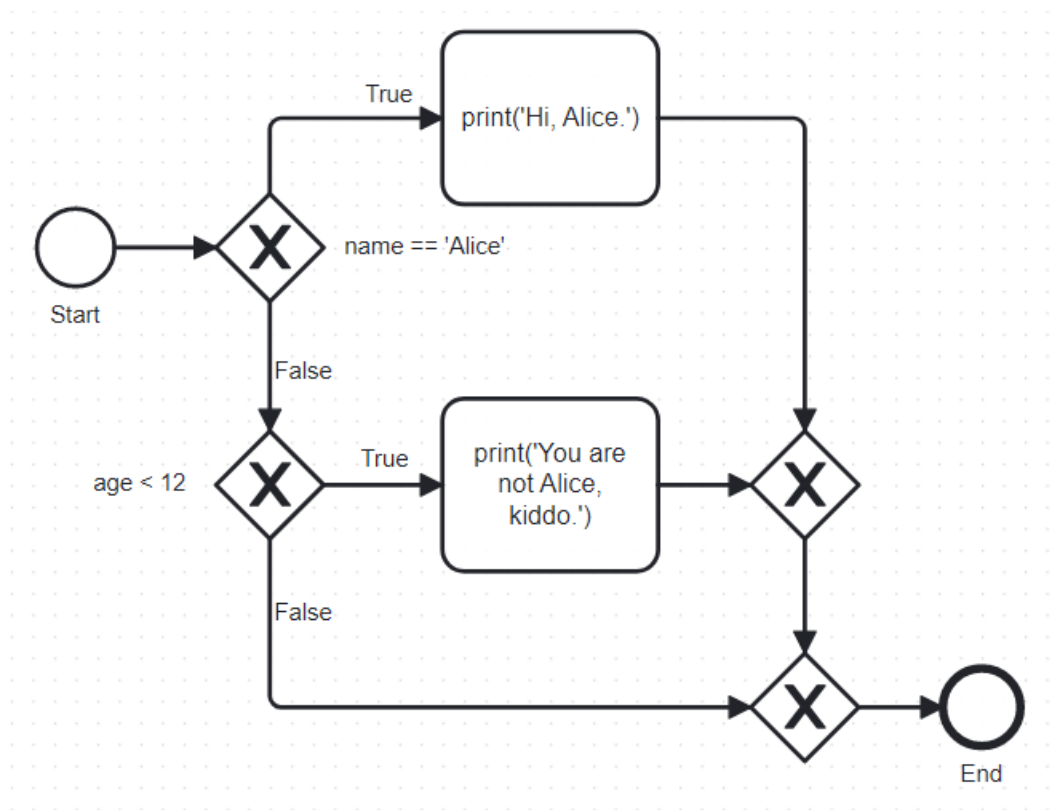


Figure 5: BPMN diagram for an `if...elif` program

The diagram shows: if neither of the conditions are fulfilled, then nothing will be printed.

- Solution (here in PythonTutor):

```
name = 'Carol'
age = 10
if name == 'Alice':
    print("Hi, Alice.")
elif age < 12:
    print("You're not Alice, kiddo.")
```

- Expand the program further in your notebook by adding two additional `elif` clauses:
  1. if `age` is greater than 2000, print "Unlike you, Alice isn't a vampire."
  2. if `age` is greater than 100, print "You are not Alice, grannie."
- Check out the program flow at [autbor.com/vampire/](http://autbor.com/vampire/).
- The BPMN diagram looks like this:
- The order of the statements does matter: once a **True** condition has been found in one of the `elif` conditions, the rest of the clauses are automatically skipped.
- Example: in [autbor.com/vampire2/](http://autbor.com/vampire2/), two of the condition/clause pairs were swapped: the first one, `age > 100` is **True** for values greater than 2000 as well - the vampire condition is a *deadlock*.
- PythonTutor can help you understand how your code works (or does not). Another useful tool is called a 'linter', after the Unix utility `lint` so named because it removes small errors from code. The VS Code IDE has a linter (I haven't tried it - low transfer value)

## 7 If, elif and else

To round off this section, bring all statements together in one program: an added `else` makes sure that at least one decision is taken.

1. in your Colab notebook, modify the 'Alice' program so that it does what the following BPMN diagram shows:
2. Test the code in your Colab notebook for `name, age = ~'Carol', 3000`.

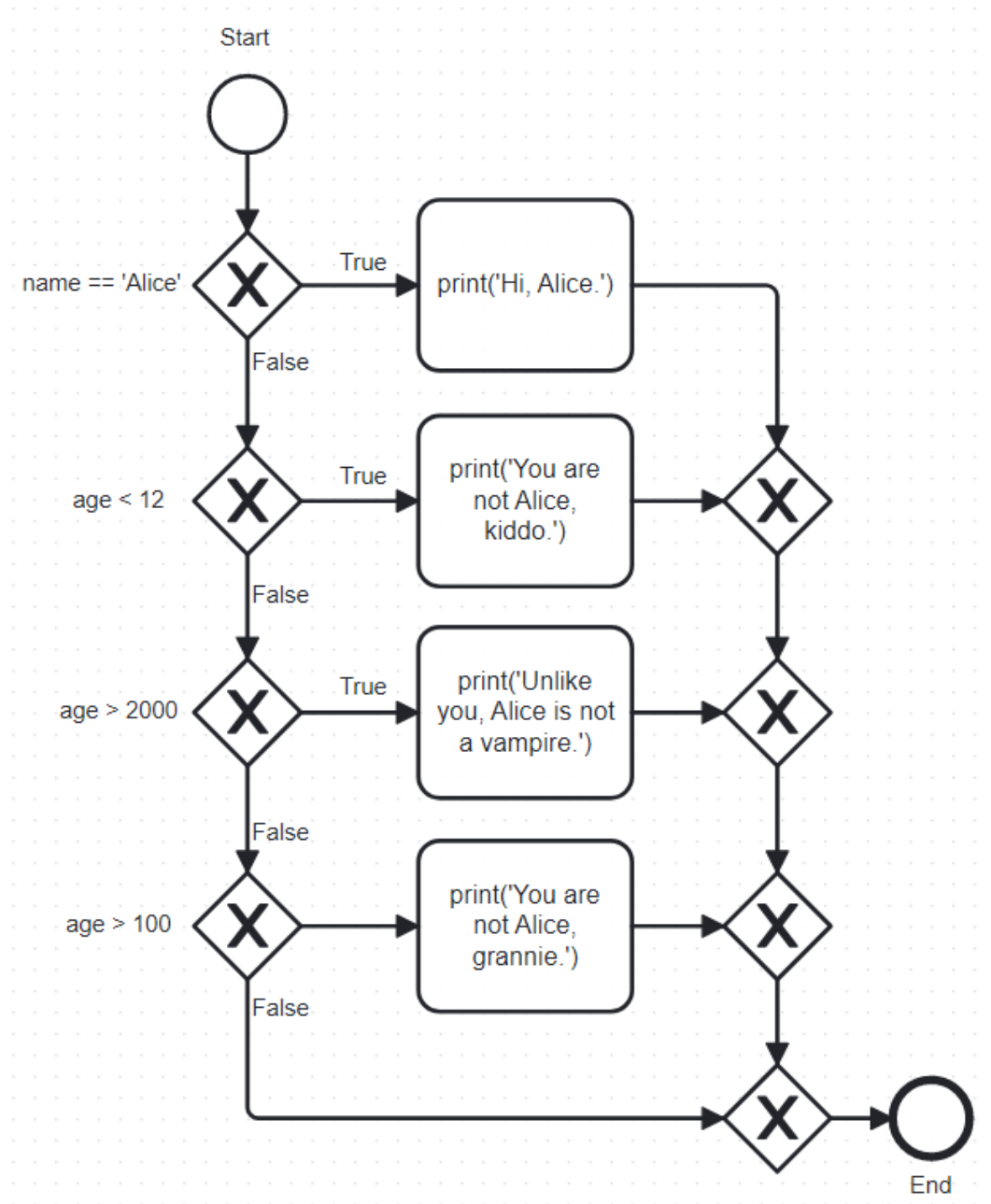


Figure 6: BPMN diagram for an expanded if...elif program

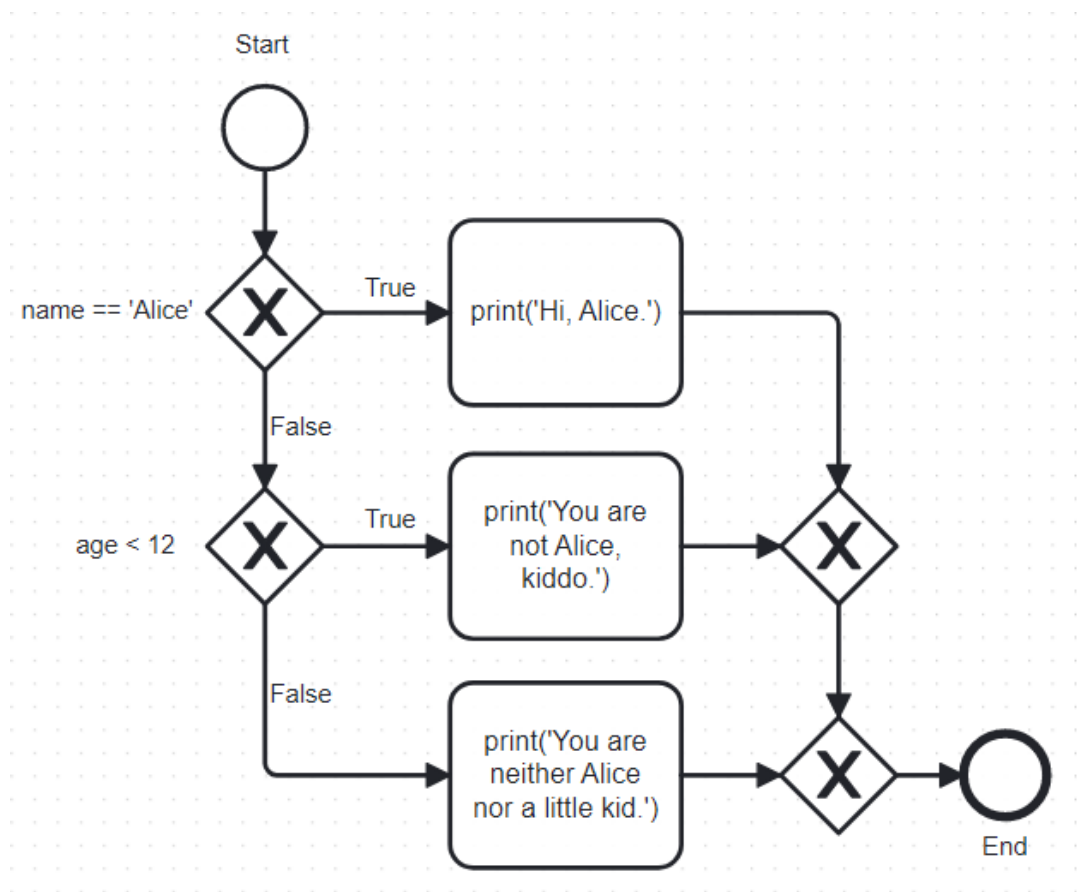


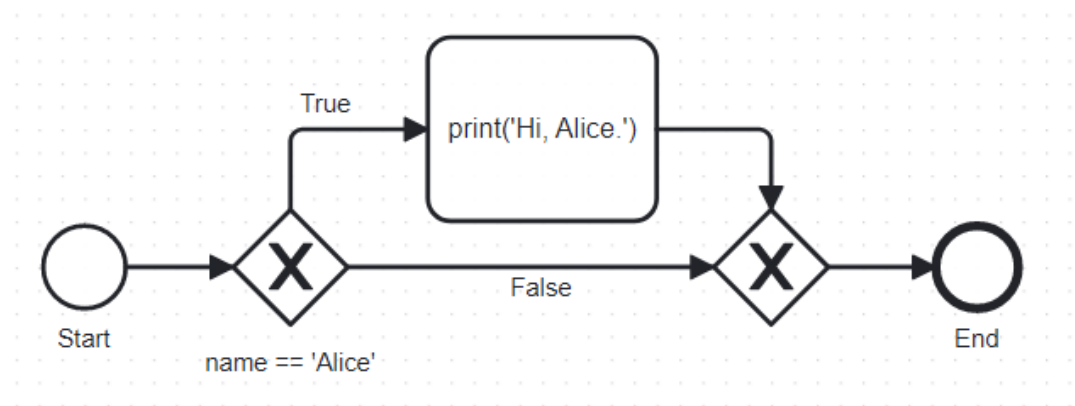
Figure 7: BPMN diagram for the 'Alice' program with if, elif, else

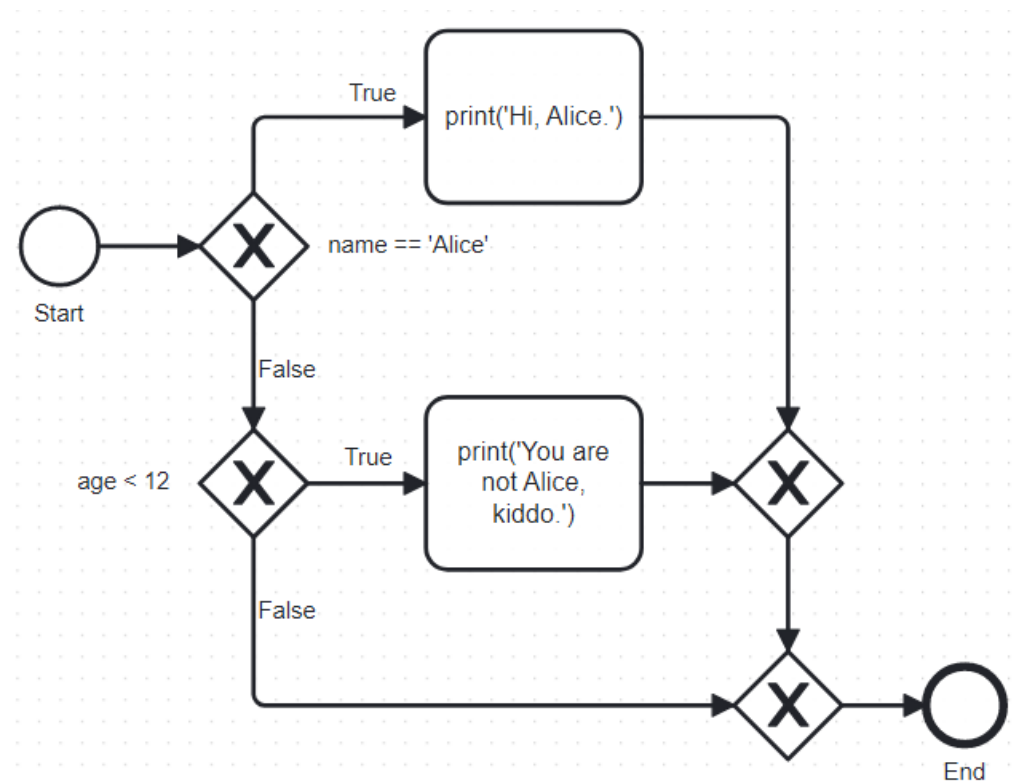
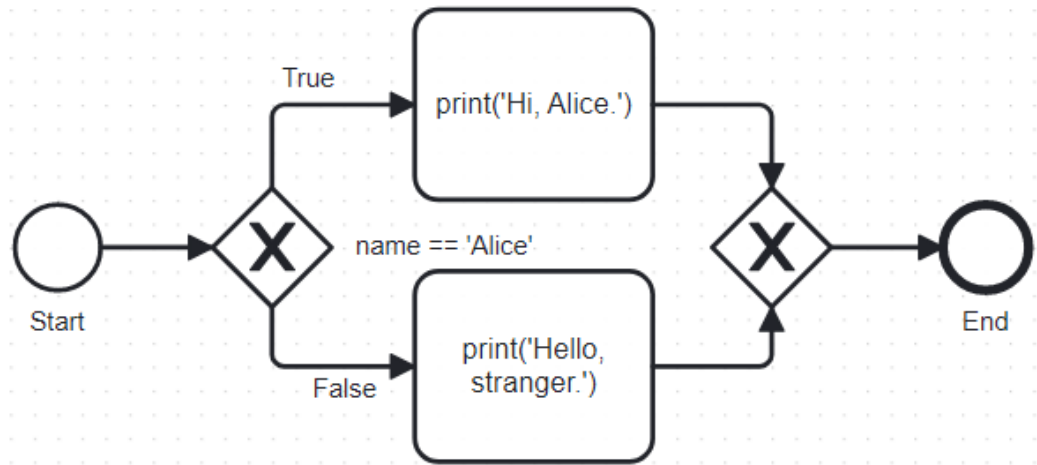
3. Copy and paste the code to PythonTutor and visualize the process.

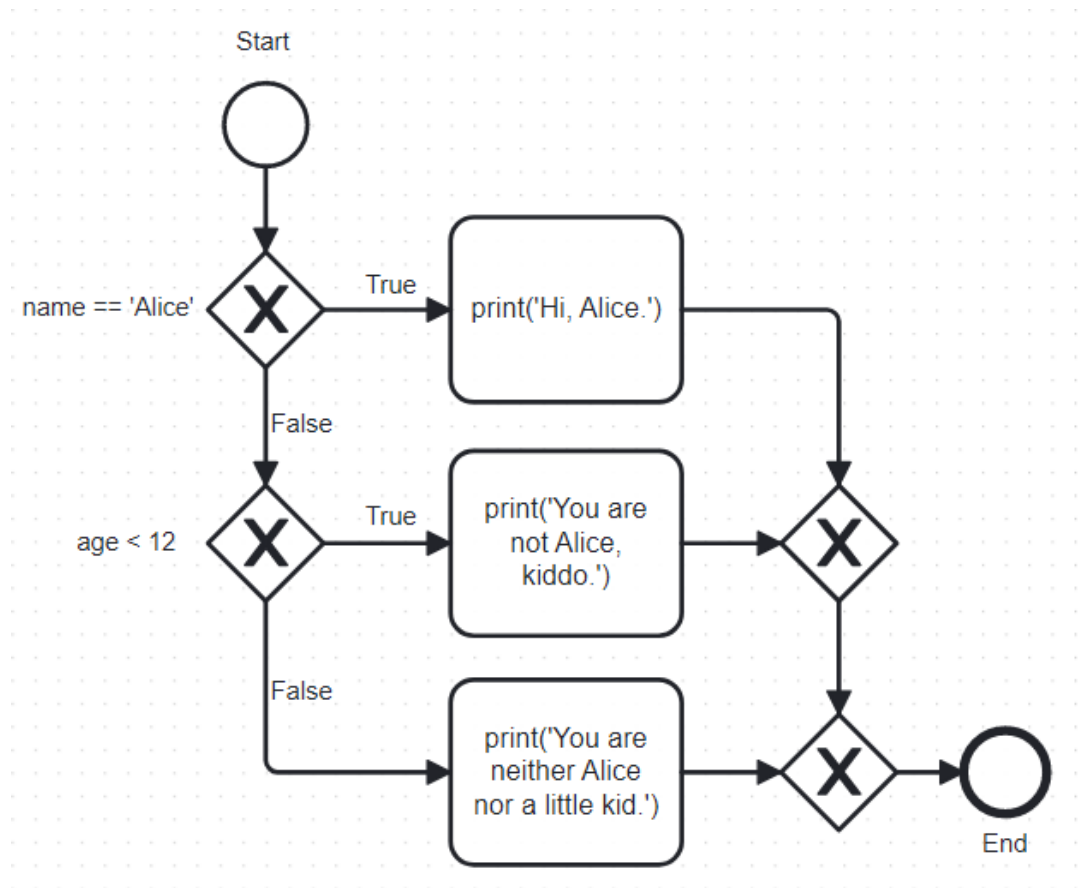
4. Solution in PythonTutor:

```
name = 'Carol'
age = 3000
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
else:
    print('You are neither Alice nor a little kid.')
```

5. All diagrams in overview:







## 8 Summary

- Decisions are modeled with conditions that evaluate to Boolean values (`True`, `False`).
- Decisions include what code to execute and what to skip (`if`, `else`, `elif`).
- Process models and flow charts show the logical structure clearly.
- You can create compound logical conditions using multiple operators, e.g. `~name == 'Alice' and int(age) > 12` to check if the input name is BOTH Alice and if she is older than 12.

## 9 Glossary

TERM/COMMAND	MEANING
<code>if</code>	keyword before a condition (start)
<code>elif</code>	keyword before a alternative condition
<code>else</code>	keyword before a final condition

## 10 References

- Sweigart, A. (2019). Automate the Boring Stuff with Python. NoStarch. URL: [automatetheboringstuff.com](http://automatetheboringstuff.com)
- Yunits, B. (2019). Which programming languages use indentation? URL: [pldb.com](http://pldb.com).