

CPS 3320 Project # 2 Write-Up

Name: Ashley Indar

Printy

1 - Central Purpose

Printy is a cross-platform software, which means that even though it was written on one platform, it can run on another. In other words, if a program was written using Printy on a Mac, it can run on a Linux or Windows system, and vice versa. The central purpose of Printy is to allow programmers to colorize and apply standard formatting to their “text with an intuitive and friendly Application Programming Interface based on flags.” Flags are essentially a signal for a function or process, and indicates the next step a program will execute. With Printy, the programmer can apply global formatting to the entire body of text, or inline formatting to specific sections or parts of text.

2 - Useful & Interesting Ways It Might Be Used w/ Two Practical Examples

One useful and interesting way Printy can be used is during web page development. When a programmer is writing code in HTML for a webpage, there are built in functions they can use to format the text how they'd like. In the same way, Printy can be used to colorize and customize sections of text at the designers discretion. This can easily dictate the flow and format of a webpage, for user readability and convenience.

Another example of a useful and interesting way Printy can be used is during game application development. Let's say for example a game developer is designing a simple crossword puzzle game. The developer can utilize Printy to make sections of the text bold, a different color, italicized, or crossed out as needed. For example, a word can be bold and red before it is found, and green and crossed out or italicized when it is found. Having the simple functions that Printy has to offer can enhance the design of a web page or application for user gratification.

3 - High- Level Overview of the Different Functions Included in the Library [as found in its official documentation - provide a link to the documentation]

To begin using Printy, the programmer must first **import Printy** in the code they are writing. Some of the different functions in Printy are as follows:

Colors

'g' -> Applies a grey color to the text

'r' -> Applies a red color to the text

'n' -> Applies a green color to the text

'y' -> Applies a yellow color to the text
'b' -> Applies a blue color to the text
'm' -> Applies a magenta color to the text
'c' -> Applies a cyan color to the text
'k' -> Applies a black color to the text
'w' -> Applies a white color to the text
'p' -> Applies the predefined color to the text

Formats

'B' -> Applies a bold font weight to the text
'D' -> Decrease the intensity, aka Dim
'U' -> Applies an underline to the text
'I' -> Applies an italic font type to the text
'H' -> Highlights the text
'S' -> crosses out the text, aka Strike

For more of its functions you can visit: <https://pypi.org/project/pretty/>

Another helpful tool is to simply install Pretty and write **from pretty import helpme** to see more functionalities and features it offers.

Tkinter

1 - Central Purpose

Tkinter is a module in the Python standard library which serves as an interface to Tk, a simple toolkit. It can be used to create a program with a Graphical User Interface, and may use a form of OO programming known as event-driven. This means that once the program is executed, it will respond to user actions. Tkinter has built in functions that allows us to build programs with widgets such as buttons, menus, and various types of entry fields and display areas. Typically these widgets are constructed in a tree structure so that each widget will have a parent widget, treeing up to the root window of the application being built. Tkinter is known as the standard GUI library for Python because it's powerful object-oriented interface to the Tk GUI toolkit provides a fast and easy way to create GUI applications.

2 - Useful & Interesting Ways It Might Be Used w/ Two Practical Examples

Programmers can use Tkinter when designing GUI applications of their desire. For example, Tkinter can be used to create a calculator which would possess both a user-interface and functionalities of a calculator. To do so, a programmer would have to import the Tkinter module, create an interface for the calculator and an input function so the user can enter their numbers to be calculated, create a function that can clear the field if the user desires, and finally provide a function that can evaluate and compute the expression to output the final result. It can also be used to create a calendar through a simple use of Tkinter methods. To do so, a programmer would have to import the Tkinter module, create the 'container' or main window which the calendar will be displayed in, add the desired amount of widgets to the main window, and apply the event trigger on the widgets. Tkinter may also be used to create fun graphical images within a window, such as a bouncing ball, or even games (for advanced programmers) such as Solitaire or Sudoku.

3 - High- Level Overview of the Different Functions Included in the Library [as found in its official documentation - provide a link to the documentation]

In order to use Tkinter, the programmer must first **import tkinter** or more commonly, **from tkinter import tk**. Some of the various basic functions in Tkinter are as follows:

`window=Tk()` #This creates a window.

`l=Label()` #This creates a label inside of the window you created, and you can even customize the text color, font, and size.

`l.place()` #This allows you to determine the position of the label within the window.

`c=Canvas()` #This allows you to draw pictures or graphics, text, and widgets.

`b=button()` #This allows you to create a button within the window.

`window.title()` #This allows you to set the title of the window as desired.

`window.mainloop()` #This is a very important function or method which allows the program to run and the window to ultimately display.

For more of its functions you can visit: <https://docs.python.org/3/library/tkinter.html>

Arrow

1 - Central Purpose

Arrow is a Python library that offers a sensible and human-friendly approach to creating, manipulating, formatting and converting dates, times and timestamps. It implements and updates the datetime type, plugging gaps in functionality and providing an intelligent module API that supports many common creation scenarios. Simply put, it helps you work with dates and times with fewer imports and a lot less code. The name was derived from the concept of the 'arrow' of time (as in the arrows that indicate the time on a mechanical clock). Arrow offers great usability because it has a fully-implemented, drop-in replacement for datetime, is time-zone aware and UTC by default, allows for timezone conversion, and formats and parses strings automatically, among many other useful features.

2 - Useful & Interesting Ways It Might Be Used w/ Two Practical Examples

Arrow provides useful ways of manipulating dates and timestamps and could be very useful to universal businesses, such as banks, that require updated dates and times of various countries for business purposes. It could also be used for personal business owners who participate in shipping or trading, and are required to stay up-to-date on universal time zones for correspondence purposes. One example of using Arrow could be creating a program that displays a window on a screen (using a library like Tkinter to create the GUI) and implementing a drop down list where the user could select which time zone they want to see, and have it displayed on the screen. The Arrow library would be imported and implemented in order to display the correct dates and times. This could be used in a business application or even from a customer point of view who needs to stay up to date on different dates and time zones for various purposes. Another useful way it can be used is in a classroom setting, where instructors who are teaching a different language can use it if a programmer implements calls `arrow.locales()` from the ArrowFactory class to display the day and month names, abbreviations and timeframes in a different language such as Azerbaijani or Danish. In the same way, a GUI can be implemented using Tkinter for the teacher to select which category they would like to be displayed to teach students from these various categories.

3 - High- Level Overview of the Different Functions Included in the Library [as found in its official documentation - provide a link to the documentation]

In order to use Arrow, the programmer must first **import arrow** or more commonly [and usefully] **from datetime import datetime**. This allows them to use full built-in functions of this library. Some of the various Arrow functions are as follows:

```
utc = arrow.utcnow() #allows the user to display the current UTC date and time
local = utc.to('US/Pacific') #allows user to display current PST date and time
```

a.year #get any datetime value

arrow.utcnow().format('YYYY-MM-DD HH:mm:ss ZZ') #allows user to display date and time in specific format [this example is in UTC time]

Classmethod **fromtimestamp**(timestamp, tzinfo=None) #constructs an **Arrow** object from a timestamp, converted to the given timezone

Arrow.locale #allows user to call different functions from various languages to display day and month names and timeframes in a different language

For more of Arrow's functions you can visit:

<https://arrow.readthedocs.io/en/latest/#regular-expressions>