

20.1 String formatting (old)



This section has been set as optional by your instructor.

©zyBooks 09/27/22 12:33 469702

Steven Cameron
WGUC859v4

Conversion specifiers

Program output commonly includes the values of variables as a part of the text. A **string formatting expression** allows a programmer to create a string with placeholders that are replaced by the values of variables. Such a placeholder is called a **conversion specifier**, and different conversion specifiers are used to perform a conversion of the given variable value to a different type when creating the string.

Conversion specifiers convert the object into the desired type. Thus, if the programmer gives a float value of 5.5 to a '%d' conversion specifier, a truncated integer value of '5' is the result.

The syntax for using a conversion specifier also includes a % symbol between the string and the value or variable to be placed into the string. Ex:

```
print('The couch is %d years old.' % couch_age)
```

The '%%' sequence displays an actual percentage sign character, as in

```
print('Annual percentage rate is %f%%' % apr).
```

PARTICIPATION
ACTIVITY

20.1.1: Using string formatting expressions.



Animation captions:

1. Simple string replacement can be done using the %s conversion specifier.
2. The %d specifier is used for integer replacement in a formatted string.
3. The %f specifier is used for float replacement in a formatted string. If an integer is passed to a floating-point specifier, the value becomes a float.

©zyBooks 09/27/22 12:33 469702

Steven Cameron
WGUC859v4

Table 20.1.1: Common conversion specifiers.

Conversion specifier(s)	Notes	Example	Output
%d	Substitute as integer.	<code>print(%d %s 09/27/22 12:33 469702 10) Steven Cameron WGUC859v4</code>	10
%f	Substitute as floating-point decimal	<code>print(%f % 15.2)</code>	15.200000
%s	Substitute as string.	<code>print(%s % 'ABC')</code>	ABC
%x, %X	Substitute as hexadecimal in lowercase (%x) or uppercase (%X).	<code>print(%x % 31)</code>	1f
%e, %E	Substitute as floating-point exponential format in lowercase (%e) or uppercase (%E).	<code>print(%E % 15.2)</code>	1.520000E+01

zyDE 20.1.1: Conversion specifiers automatically convert values.

The program below prints the average payday loan interest rate of 410.9 percent (not see [Wikipedia: Payday loan](#)). Try inputting the integer 411 instead, noting how the type converted by %f to 411.0.

```
Load default template...  
1 apr = float(input('Enter APR:\n'))  
2  
3 # Print using a float conversion specifier  
4 print('Annual percentage rate as a float')  
5  
6 # Print using a decimal conversion specifier  
7 print('Annual percentage rate as an int')  
8
```

410.9
©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Run

PARTICIPATION ACTIVITY

20.1.2: String formatting.



Complete the code using formatting specifiers to generate the described output.

- 1) Assume price = 150.



I saved \$150!

```
print('I saved $' + '!' + '%  
price)
```

Check

Show answer

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

- 2) Assume percent = 40.



Buy now! Save 40%!

```
print('Buy now! Save' + '!' + '%  
percent)
```

Check

Show answer

Multiple conversion specifiers

Multiple conversion specifiers can appear within the string formatting expression. Expressions that contain more than one conversion specifier must specify the values within a tuple following the '%' character. The following print statement will print a sentence including two numeric values, indicated by the conversion specifiers %d and %f.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.1.1: Multiple conversion specifiers.

```
years = 15
total = 500 * (years * 0.02)
print('Savings after %d years is: %f' % (years,
total))
```

Savings after 15 years is:
150.000000

PARTICIPATION ACTIVITY

20.1.3: Multiple conversion specifiers.



Complete the code using formatting specifiers to generate the described output. Use the indicated variables.

- 1) Assume `item = 'burrito'`
and `price = 5`.



The burrito is \$5

```
print('The  is $%d' %
(item, price))
```

Check

[Show answer](#)

- 2) Assume `item = 'backpack'`
and `weight = 5.2`.



The backpack is 5.200000 pounds.

```
print('The %s is 
pounds.' % (item, weight))
```

Check

[Show answer](#)

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4



- 3) Assume `city = 'Boston'` and
`distance = 2100.`

We are 2100 miles from Boston.

```
print('We are %d miles from  
%s.' % (  
    ))
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Check

[Show answer](#)

**CHALLENGE
ACTIVITY**

20.1.1: Printing a string.



Write a *single* statement to print: `user_word,user_number`. Note that there is no space between the comma and `user_number`.

Sample output with inputs: 'Amy' 5

Amy,5

334598.939404.qx3zqy7

```
1 user_word = str(input())  
2 user_number = int(input())  
3  
4 ''' Your solution goes here '''  
5
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Run

**CHALLENGE
ACTIVITY**

20.1.2: String formatting.



334598.939404.qx3zqy7

Start

©zyBooks 09/27/22 12:33 469702

Steven Cameron

Select the most appropriate conversion specifiers. JC859v4

num_gifts = 12

```
print('For Christmas, I got %d' % num_gifts)
```

For Christmas, I got 12 gifts from my family.

1

2

3

Check**Next**

20.2 String formatting using dictionaries



This section has been set as optional by your instructor.

©zyBooks 09/27/22 12:33 469702

Steven Cameron
WGUC859v4

Mapping keys

Sometimes a string contains many conversion specifiers. Such strings can be hard to read and understand. Furthermore, the programmer must be careful with the ordering of the tuple values, lest items are mistakenly swapped. A dictionary may be used in place of a tuple on the right side of the conversion operator to enhance clarity at the expense of brevity. If a dictionary is used, then all

conversion specifiers must include a **mapping key** component. A mapping key is specified by indicating the key of the relevant value in the dictionary within parentheses.

PARTICIPATION ACTIVITY

20.2.1: Using a dictionary and conversion specifiers with mapping keys.

**Animation captions:**

©zyBooks 09/27/22 12:33 469702
Steven Cameron

1. A mapping key is specified by indicating the key of the relevant value in the dict within parentheses.

Figure 20.2.1: Comparing conversion operations using tuples and dicts.

```
import time
gmt = time.gmtime() # Get current Greenwich Mean Time

print('Time is: {:02d}/{:02d}/{:04d} {:02d}:{:02d} {:02d} sec' \
      .format(gmt.tm_mon, gmt.tm_mday, gmt.tm_year, gmt.tm_hour, gmt.tm_min,
              gmt.tm_sec))
```

```
Time is: 06/07/2013 20:16 24 sec
...
Time is: 06/07/2013 20:16 28 sec
```

```
import time
gmt = time.gmtime() # Get current Greenwich Mean Time

print('Time is: %(month)02d/%(day)02d/%(year)04d %(hour)02d:%(min)02d %(sec)02d sec' %
      {
          'year': gmt.tm_year, 'month': gmt.tm_mon, 'day': gmt.tm_mday,
          'hour': gmt.tm_hour, 'min': gmt.tm_min, 'sec': gmt.tm_sec
      })
```

```
Time is: 06/07/2013 20:16 24 sec
...
Time is: 06/07/2013 20:16 28 sec
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

PARTICIPATION ACTIVITY

20.2.2: Mapping keys.



Complete the print statement to produce the given output using mapping keys.

- 1) "I need 12 lilies, 6 roses, and 18



```
tulips."  
print ('I need %(lilies)d  
lilies, %(roses)d roses, and  
%(tulips)d tulips.' % {  
    } )
```

CheckShow answer

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4



- 2) "My name is Jerome and I'm 15 years old."

```
print ('My name is %(name)s  
and I am %(age)d years old' %  
{  
    } )
```

CheckShow answer

20.3 Basic graphics



This section has been set as optional by your instructor.

Creating a graphics frame

Python supports a set of objects for developing graphical applications. A **graphical application** is a program that displays drawings and other graphical objects. **TKinter** is a standard Python package for graphical applications. TKinter displays contents inside a window called a **frame** using a **Frame** object. The following program shows how to create and configure an Frame object to display an empty application window.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.3.1: Creating a Frame object for a graphical application.

```
import tkinter as tk

class Application(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master

        # Set the frame's title
        self.master.title('An Empty Frame')
        self.pack()

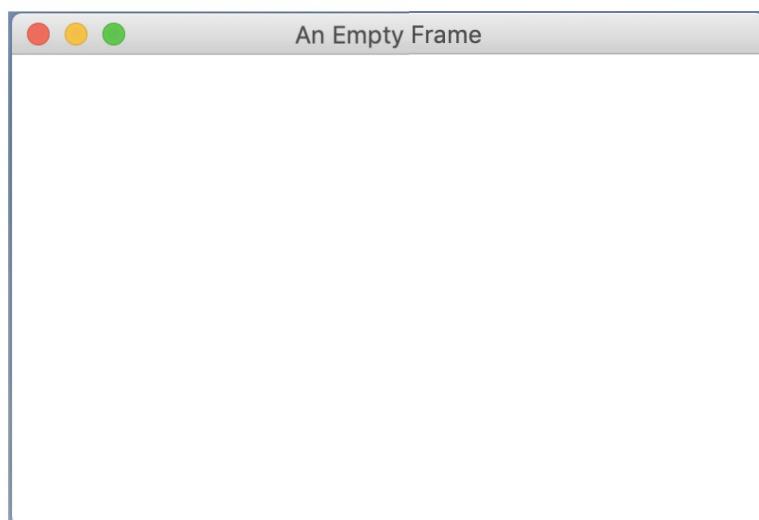
app_frame = tk.Tk()

# Set the frame's width (400) and height (250) in
# pixels
app_frame.geometry('400x250')

# Make the frame visible to the user
app = Application(master=app_frame)
app.mainloop()
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.3.2: Screenshot of empty application window.



©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Constructing a Frame object does not immediately display a frame. The program uses the methods supported by the Frame object to configure and display the frame as follows:

1. **Set the frame's size** by calling the `geometry()` method with arguments for the width and height, as in `app_frame.geometry('400x250')`. Forgetting to set the frame's size results in a frame too small to see.

2. **Set the frame's title** by calling the title() method with a String as the argument.

3. **Make the frame visible** to the user by calling the mainloop() method.

PARTICIPATION
ACTIVITY

20.3.1: Configuring a Frame.



Select the code statement that would resolve the described problem. Assume an empty Frame object named appFrame.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

- 1) The frame window lacks a title. User would like the title to be "My program".

- self.master.title(My program)
- self.master.title('My program')

- 2) The program called the mainloop() method correctly, but the frame is not visible on the screen. The frame should be 500 pixels wide and 300 pixels tall.

- app_frame.geometry('500x300')
- app.mainloop(false);
- app_frame.geometry('300x500')



Drawing graphical objects

A Frame can be used to draw graphical objects, such as rectangles, circles, and lines. To display graphical objects, a programmer can add a Canvas object to a frame. A **Canvas** is a graphical component that a programmer can use to draw basic shapes.

The following program demonstrates how to build a class that creates a Canvas to draw 2D graphics.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.3.3: Basic example showing how to create a class to draw 2D graphics using Canvas.

```
import tkinter as tk
from tkinter import Canvas, Frame,
BOTH

class Application(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.pack(fill=BOTH, expand=1)

    canvas = Canvas(self)
    # Write your drawing
instructions

app_frame = tk.TK()
app_frame.geometry('400x250')
app = Application(master=app_frame)
app.mainloop()
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

The above code defines a class named Application that uses a Canvas object. A programmer completes the template by providing custom drawing instructions after the Canvas object has been created. In the animation below, the programmer uses Canvas's create_rectangle() to draw a rectangle in the frame.

Canvas's create_rectangle() takes the following arguments:

1. **Arguments 1 and 2:** coordinate for the top left corner (x0, y0) of the rectangle
2. **Arguments 3 and 4:** coordinate for the bottom right corner (x1, y1)
3. **Argument 5 (optional):** rectangle outline color (if not set, the outline will be black)
4. **Argument 6 (optional):** rectangle fill color

Many more optional arguments exist for the create_rectangle() method, such as width (width of border) and dash (make the border dashed).

PARTICIPATION ACTIVITY

20.3.2: Drawing a filled rectangle.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4



Animation content:

undefined

Animation captions:

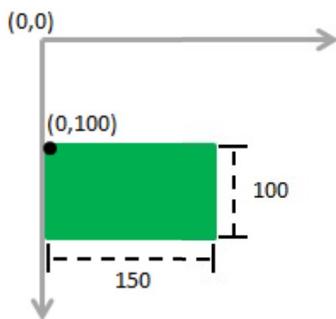
1. The canvas variable is set to a Canvas object for drawing 2D graphics.

2. The `create_rectangle()` method draws a rectangle with the coordinates (10, 75) and (150, 50), with the outline and fill color set to "cyan".

The programmer needs to know the positioning coordinate system in order to draw shapes in the intended location. As the following figure illustrates, the top-left corner of a Canvas corresponds to coordinates (0, 0). The x-coordinate increases horizontally to the right and the y-coordinate increases vertically downward.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.3.4: Graphics coordinate system.



PARTICIPATION ACTIVITY

20.3.3: Drawing colored rectangles.



Select the code statement that would resolve the described problem. Assume `canvas` is a `Canvas` object.

- 1) The user wants a rectangle's top left corner coordinates to be `x0=5` and `y0=5` and the rectangle to be `100 x 100` pixels in size.



- `canvas.create_rectangle(5, 5, 105, 105)`
- `canvas.create_rectangle(5, 100, 100, 100)`
- `canvas.create_rectangle(0, 0, 100, 100)`

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

- 2) The user wants a pink rectangle to have a blue outline.



- canvas.create_rectangle(10,
10, 200, 50,
outline='pink',
fill='blue')
- canvas.create_rectangle(10,
10, 200, 50,
outline='blue',
fill='pink')
- canvas.create_rectangle(10,
10, 200, 50,
outline='blue')

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Ex: Drawing a basic histogram

The following program uses a Canvas object to draw a simple histogram using Canvas's create_rectangle() method. The program first creates a HistogramViewer object named histogram_viewer and adds the object to the Frame. The HistogramViewer class creates three rectangles by calling Canvas's create_rectangle() method.

Figure 20.3.5: Drawing a histogram in a frame.

HistogramApp.py

```
import tkinter as tk
from tkinter import Canvas, Frame, BOTH

class HistogramViewer(Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.master.title('Histogram Viewer')
        self.pack(fill=BOTH, expand=1)

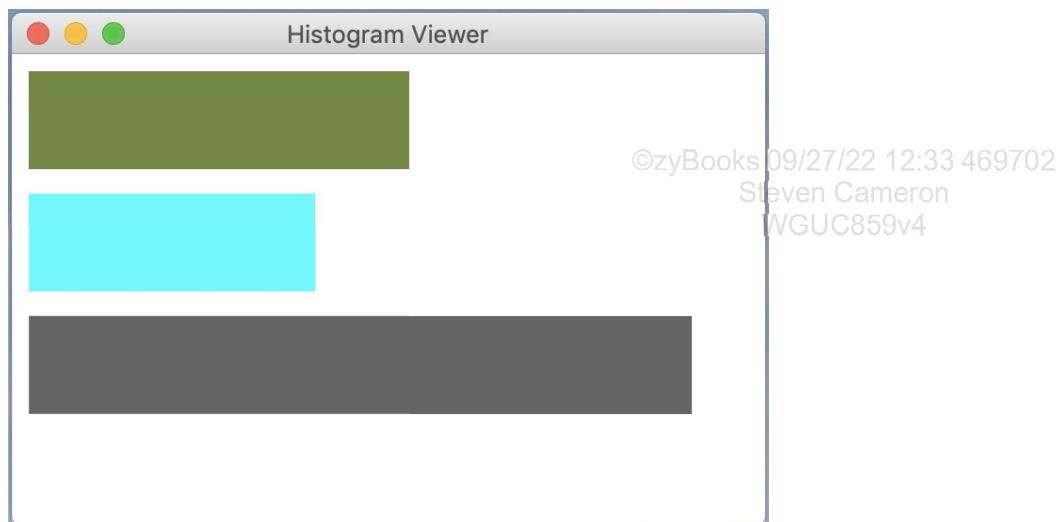
        canvas = Canvas(self)
        canvas.create_rectangle(10, 10, 210, 60, outline='darkolivegreen4',
        fill='darkolivegreen4')
        canvas.create_rectangle(10, 75, 160, 125, outline='cyan', fill='cyan')
        canvas.create_rectangle(10, 140, 360, 190, outline='gray40', fill='gray40')

        canvas.pack(fill=BOTH, expand=1)
        self.pack()

app_frame = tk.Tk()
app_frame.geometry('400x250')
histogram_viewer = HistogramViewer(master=app_frame)
histogram_viewer.mainloop()
```

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Figure 20.3.6: Screenshot of HistogramViewer application.



**PARTICIPATION
ACTIVITY**

20.3.4: Drawing rectangles.



Which code segment (type the number) performs the described operation? Assume the Canvas object is called graphicsObj.

1. `graphicsObj.create_rectangle(0, 0, 150, 100, fill='green')`
2. `graphicsObj.create_rectangle(0, 100, 200, 300, outline='red', fill='red')`
3. `graphicsObj.create_rectangle(0, 100, 50, 250, outline='purple', fill='purple')`

1) Draws a filled in square.



Check

[Show answer](#)

2) Draws the outline of a rectangle



50 pixels wide and 150 pixels in height.

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Check

[Show answer](#)

3) Draws a rectangle whose top-left corner is located at the



origin of the coordinate system.

Check**Show answer**

- 4) Draws a rectangle with a black outline.



©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4

Check**Show answer**

Canvas provides methods for drawing structured graphics, of which some common shapes are summarized below:

Table 20.3.1: Summary of common shapes for drawing.

Shape	Description	Documentation
Rectangle	Draws a rectangle on the canvas.	create_rectangle() method
Oval	Draws an ellipse on the canvas.	create_oval() method
Line	Draws a line on the canvas.	create_line() method
Polygon	Draws a polygon on the canvas.	create_polygon() method

Exploring further:

- [Color chart for TKinter \(all available color names\)](#)

©zyBooks 09/27/22 12:33 469702
Steven Cameron
WGUC859v4