Python Data Analysis Basics

Syntax

STRING FORMATTING AND FORMAT SPECIFICATIONS

- Insert a value intro a string:

```
world_cup_semifinal = "The final score was Croatia {}, England {}".format(2,1)
```

- Format specification for two decimal places:

```
two_decimal_places = "I own {:.2f}% of the company".format(32.5548651132)
```

-Format specification for comma separator:

```
india_pop = The approximate population of {} is {}".format("India",1324000000)
```

- Format specification for padding and alignment:

```
left_aligned_padded_20_chars   = "| {:<20} |".format("MY_VALUE")
center_aligned_padded_20_chars = "| {:^20} |".format("MY_VALUE")
right_aligned_padded_20_chars  = "| {:>20} |".format("MY_VALUE")
```

- Format specification to derive padding width from a variable:

```
left_aligned  = "| {:<{}} |".format("MY_VALUE", padding_width)
```

- Order for format specification:

```
balance_string = "Your bank balance is {:>20,.2f}"].format(12345.678)
```

Concepts

- The `str.format()` method allows you to insert values into strings without explicitly converting them.
- The `str.format()` method also accepts optional format specifications which you can use to format values so they are more easily read.

Resources

- Python Documentation: Format Specifications
- PyFormat: Python String Formatting Reference