Adv. Strings

David H Smith IV

University of Illinois Urbana-Champaign

Tues, Oct 4 2021

Reminders

Reminders

- Post-reading 9p1 is due Tommorow (will be posted after class).
- Homework 8 is due tommorow.
- Participation 9p1 is due Tommorow.
- Lab 4 is due Sunday after next.

Strings Slicing

Poll Question: Slicing

What is the result of running this code?

```python
my_str = "CS 105"
print(my_str[1:2])
```

- A 'C'
- B 'CS'
- C 'CS '
- D 'S'
- E 'S '

Poll Question: Slicing

What is the result of running this code?

```python
my_str = "CS 105"
print(my_str[-4:-2])
```

- A ’S 1’
- B ’S 10’
- C ’ 1’
- D ’ 10’

Poll Question: Slicing

What is the result of running this code?

```python
my_str = "CS 105"
print(my_str([::2]))
```

- A ’C’
- B ’CS’
- C ’S’
- D ’C 0’

Slicing

- A `string[start:stop:interval]`
- B Like range, start is inclusive stop is exclusive.
- C Interval default is 1
- D Interval is optional

Split

Poll Question: Splitting

What is the result of running this code?

```python
my_str = "CS 105 rox"
result = my_str.split()
```

- A `("CS", "105", "rox")`
- B `["CS 105 rox"]`
- C `["CS", "105 rox"]`
- D `["CS", "105", "rox"]`

Poll Question: Splitting

What is the result of running this code?

```
csv = "1, 2, 3, 4"
result = csv.split(",")
```

- A `['1']`
- B `['1, 2, 3, 4']`
- C `['1', '2', '3', '4']`
- D `['1,', '2,', '3,', '4']`

Join

Poll Question: Joining

What is the result of running this code?

```
numlist = [1, 2, 3, 4]
result = ",".join(numlist)
```

- A `'1234'`
- B `'1,2,3,4'`
- C `'1, 2, 3, 4'`
- D `TypeError`

Poll Question: Joining

What is the result of running this code?

```python
numlist = [1, 2, 3, 4]
result = ",".join(numlist)
```

- A '1234'
- B '1,2,3,4'
- C '1, 2, 3, 4'
- D TypeError

How do we fix this?

A Common Pattern

The generic pattern:

```
mylist = input_data.split(<separator>)
... data processing ...
outputstring "<separator>".join(my_list)
```

An example of this being done on one line:

```
output = ",".join(string.split(",")[::2])
```

Pattern Practice

Write some code that takes a string with comma separated integers that converts the string into the square of each original value.

Go to PrairieLearn to do this problem

Pattern Practice

```python
def foo(numlist):
  squaredlist = []
  for num in numlist.split(","):
    squaredlist.append(str(int(num) ** 2))
  return squared_csv = ",".join(squaredlist)
```

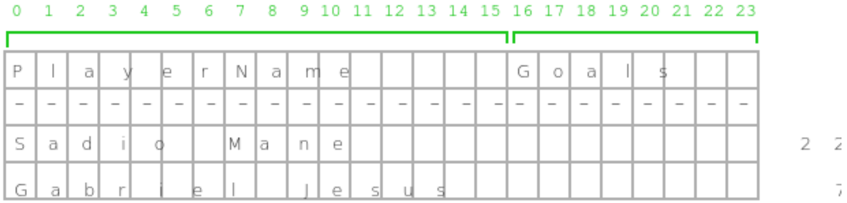
Adv. String Formatting

Adv. String Formatting

```
format_string = '{name:16}{goals:8}'

print(format_string.format(name='Player Name', goals='Goals'))
print('-' * 24)

print(format_string.format(name='Sadio Mane', goals=22))
print(format_string.format(name='Gabriel Jesus', goals=7))
```



```
Player Name     Goals
------------------------
Sadio Mane            22
Gabriel Jesus          7
```

Alignment

For a field width of 10:

1. **Left-aligned:** `"{:<10}".format(x)`
2. **Right-aligned:** `"{:>10}".format(x)`
3. **Centered:** `"{:^10}".format(x)`

Notice the similarity between field width and how we set the number of decimals after a floating point: `"{:.2f}".format(math.pi)` $\rightarrow$3.14.

Alignment

For a field width of 10:

1. **Left-aligned:** `"{:<10}".format(x)`
2. **Right-aligned:** `"{:>10}".format(x)`
3. **Centered:** `"{:^10}".format(x)`

Notice the similarity between field width and how we set the number of decimals after a floating point: `"{:.2f}".format(math.pi)` $\rightarrow$3.14.

We can use a fill character to consume any unused spaces in the field width:

1. **Left-aligned:** `"{:-<10}".format(x)`
2. **Right-aligned:** `"{:->10}".format(x)`
3. **Centered:** `"{:-^10}".format(x)`

Formatting Practice

Create a function that takes a list of lists where each sub list contains 4 elements. Create and return a new list of strings where each string is composed of the four elements in each sublist and:

1. the first element is center aligned with a field with of 10
2. the second element is right aligned with a field width of 8
3. the third element is left aligned with a field width of 9
4. the fourth element is center aligned with a field width of 10 and the filler character "-".

Problem is on PrairieLearn

Example Function Call

```python
x = [
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"]
]
formatted_x = formatted_str_list(x)
```

Pattern Practice

```python
def formatted_str_list(x):
  formatted_strs = []
  for a, b, c, d in x:
    x = "{:^10}{:>8}{:<9}{:-^10}".format(a, b, c, d)
    formatted_strs.append(x)
  return formatted_strs

x = [
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"],
  ["This", "is", "a", "list"]
]
formatted_x = formatted_str_list(x)
```

Patterns (Part 1)

Counting Pattern

```
def count(collection):
  counter = 0
  for item in collection:
    if <item meets condition>:
      counter += 1
  return counter
```

Computing a Sum/Total

```python
def sum(collection):
  total = 0
  for item in collection:
    total += item
  return total
```

Pattern Practice

Spend remaining class time working on last three problems in PrairieLearn.