Interactive Programs

- Strings can have placeholders, the values can follow:
  ```
  print 'Hi%s' % 'Markus'
  ```
- and we can do that with variables, too
  ```
  greeting = 'Hi%s'
  name = 'Markus'
  print greeting % name
  ```
- and with more than one value
  ```
  greeting = 'Hi%s, what\'s \today, it\'s %i\ degrees outside.'
  vals = ('Markus', 'cold', 36)
  print greeting % vals
  ```
- Make sure to use a tuple, not a list, for the values!
Width, Precision, and Zeroes

- in displaying a number, we define the width and precision:

```python
from math import pi
print('Pi: %.10.3f' % pi)
```

- a - creates left alignment, a + adds the plus sign to the number

```python
from math import pi
print('Pi: %−10.3f' % pi)
```

- and we can fill the whole thing with zeroes

```python
from math import pi
print('Pi: %.010.3f' % pi)
```

String and Tuples

Example

You can format tuples all at once, e.g.

```python
's: %4.2f % ('pi', pi)
```

Example

1. Find where `in` starts in the phrase `needle in a haystack`
   phrase.find('in')

2. If `needle in a haystack` contains `hay`, print `hey`
   if phrase.find('hay')>=0:
     print('hey')

Example

1. Find where `in` starts in the phrase `needle in a haystack`
   phrase.find('in')

2. If `needle in a haystack` contains `hay`, print `hey`
   if phrase.find('hay')>=0:
     print('hey')

- find does NOT return a Boolean value: if it does not find the substring, it returns -1
Join and Split

1. Split the haystack phrase into multiple words
   ```python
   words=phrase.split()
   ```

   2. Reverse the order of the words
   ```python
   words.reverse()
   ```

   3. Join the words back together with commas
   ```python
   ','.join(words)
   ```

Changing Case

1. Make ALLCAPS all lowercase
   ```python
   'ALLCAPS'.lower()
   ```

2. Make all but the first letter of ALLCAPS lowercase
   ```python
   'ALLCAPS'.title()
   ```

Replace

1. Replace needle with noodle in the haystack phrase
   ```python
   phrase.replace('needle', 'noodle')
   ```
Replace

1. Replace *needle* with *noodle* in the haystack phrase
   `phrase.replace('needle', 'noodle')`
2. Replace *e* with *o* in the haystack phrase
   `phrase.replace('e', 'o')`

Strip

1. Strip off newline characters from end of the haystack phrase
   `phrase=phrase.strip()`
2. Strip off any leading or trailing whitespace from the haystack phrase, and convert to upper case
   `phrase=phrase.strip().upper()`