SKILLPILLS

Skill Pill: Programming with Python

String manipulation

Jeremie Gillet

OIST

Strings

- Defined with quotes (single or double)
  - `>>>> s = "Skill Pills are awesome!"`
- Single characters are strings too
- Printing several strings
  - `>>>> print "I'm telling you," , s`
    - `I'm telling you, Skill Pills are awesome!`
- Special characters: '\n' (new line), '\t' (tabulation)

Treating strings as lists

- >>> s = "Skill Pills are awesome!”
- >>> "Skills" in
  - ★ False
- >>> "boring" not in s
  - ★ True
- >>> s[0]
  - ★ 'S'
- >>> s[0:11]
  - ★ 'Skill Pills'
- >>> "I love " + s[0:11]
  - ★ 'I love Skill Pills'
- >>> 3 * "Hip hip? Hooray! "
  - ★ 'Hip hip? Hooray! Hip hip? Hooray! Hip hip? Hooray! '
- >>> len(s)
  - ★ 24
- >>> [min(s), max(s)]
  - ★ [' ', ‘w']
- >>> s.index("awesome")
  - ★ 16
- >>> [s.count("a"), s.count("S"), s.count("s")]
  - ★ [2, 1, 2]

Formatting

SKILLPILLS

- >>> "I like %s and %s." % ("apples", “oranges") **# Old style**
  - 'I like apples and oranges.’
- >>> "I like {} and {}." .format("apples", “oranges") **# New style**
  - 'I like apples and oranges.’
- >>> "I like {1} and {0}." .format("apples", "oranges")
  - 'I like oranges and apples.’
- >>> "I like {1}, {0} and {0}." .format("apples", "oranges")
  - 'I like oranges, apples and apples.’

Padding and truncating

SKILLPILLS

- >>> "Take {:9} Skill Pills".format('three') # Padding right
  - 'Take three Skill Pills'
- >>> "Take {:>9} Skill Pills".format('three') # Padding left
  - 'Take three Skill Pills'
- >>> "Take {:{}} Skill Pills".format('three', 9) # Padding right
  - 'Take three Skill Pills'
- >>> "Take {:^9} Skill Pills".format('three') # Centering
  - 'Take three Skill Pills’
- >>> "Take {:=^9} Skill Pills".format('three') # Centering between =
  - 'Take ==three== Skill Pills’
- >>> "Take {:.5} Skill Pills".format('three million') # Truncating
  - 'Take three Skill Pills'

Formatting numbers

SKILLPILLS

- >>> “1+1= {}".format(3.141592653)
  - ‘1+1= 3.141592653'
- >>> “1+1= {:f}".format(3.141592653)
  - ‘1+1= **3.141593**’
- >>> “1+1= {:f}".format(1.5)
  - ‘1+1= **1.500000**'
- >>> “1+1= {:06.4}”.format(3.141592653)
  - ‘1+1= **03.142**'
- >>> “1+1= {:d}".format(2)
  - '1+1= 2'
- >>> “1+1= {:05d}".format(2)
  - '1+1= **00002**'
- >>> "1+1= {:+d}".format(6)
  - '1+1= **+6**'
- >>> "1+1= {: d}".format(6)
  - '1+1= **6**'

Other possibilities

SKILLPILLS

- >>> "I like **{fruit}** and **{bestfruit}**." .format(**fruit=**"apples", **bestfruit=**"oranges")
  - ★ 'I like apples and oranges.'
- >>> **fruits**={'**fruit**':'**apples**','**bestfruit**':'**oranges**'} # Dictionary
- >>> "I like **{fruit}** and **{bestfruit}**.".format(****fruits**)
  - ★ 'I like apples and oranges.'
- >>> "I like **{p[fruit]}** and **{p[bestfruit]}**.".format(**p=fruits**)
  - ★ 'I like apples and oranges.'
- >>> from datetime import datetime
- >>> '{:%Y-%m-%d %H:%M}'.format(datetime(2001, 2, 3, 4, 5))
  - ★ '2001-02-03 04:05'

String methods

- `>>> s="skill pills"`
- `>>> s.capitalize()`
  - ★ `'Skill pills'`
- `>>> s.title()`
  - ★ `'Skill Pills'`
- `>>> s.upper()`
  - ★ `'SKILL PILLS'`
- `>>> s.islower()`
  - ★ `True`
- `>>> s.isupper()`
  - ★ `False`
- `>>> "31415".isdigit()`
  - ★ `True`
- `>>> " + ".join(["1","2","five","70","..."])`
  - ★ `'1 + 2 + five + 70 + …'`
- `>>> "  hi\n ".strip()`
  - ★ `'hi'`
- `>>> s.split()`
  - ★ `['skill', 'pills']`
- `>>> s.find("pills")`
  - ★ `6`
- `>>> s.replace("pills","syringes")`
  - ★ `'skill syringes'`
- `>>> "1".zfill(3)`
  - ★ `'001'`

Exercices

- Go to http://rosalind.info/
- Register
- First: Python village
- Second: Bioinformatics Stronghold