PRACTICAL 2

PLAY WITH FUNCTIONS

Presented by Qing Lyu / 吕晴

Adapted from Fall 2017 tutorial

Q&A - HW0

- Has everyone got Python, NLTK and PyCharm (or another IDE) installed?
- Any difficulty?

WHAT WE'VE TALKED ABOUT LAST TIME

- Why Python
- Basic data types in Python
- Sequence operations (any Q?)

A quick test:

- `my_sent = ['I','love','NLP']`
- `print(my_sent[1][:4])`

What's the output?

- `love`

GETTING TO KNOW FUNCTION

WHAT & WHY

- What's function?
  - Same as you understand it: f(x)=y
  - x: input, y: output
- Why function?

```
r1 = 12.34
r2 = 9.08
r3 = 73.1
......
s1 = 3.14 * r1 * r1
s2 = 3.14 * r2 * r2
s3 = 3.14 * r3 * r3
......
```

Now, what if I want to change 3.14 to 3.1415926?

```
s1 = area_of_circle(r1)
```

WHAT DOES A FUNCTION LOOK LIKE?

- try:

```
def square(x):
    return x * x
```

  - **def:** means ‘define’, tells Python you’re starting a function
  - `square:` function name
  - `x`: a parameter, like the x in f(x)=y
  - `x * x`: the value that the function returns, like the y in f(x)=y

- try:

```
print(square(5))
```

  - We’re calling/调用 the function we’ve defined above!

WHAT DOES A FUNCTION LOOK LIKE?

- Note:
  - Codes inside a function should be indented/缩进 (i.e. having a “tab” before). Pycharm automatically does this for you. In Python, spaces are meaningful!
  - A function can have 0,1, 2… parameter(s) and return value(s)!

YOUR TURN

- What does this function do?

```
def mystery_func(x, n):
  result = 1
  while n > 0:
    n -= 1
    result *= x
  return result
```

- i.e. what is the output of `print(mystery_func(3, 3))` ?
- Loop / 循环语句： we'll talk about it in detail next time

YOUR TURN

- Fibonacci Series/斐波那契数列: 1, 1, 2, 3, 5, 8, 13…
- Complete the following function for calculating the nth item of Fibonacci Series:

```python
def fib_nth(n):
    a, b = 0, 1
    while n > 0:
        a, b = b, a + b
        n -= 1
    return a
```

INBUILT FUNCTIONS / 内置函数

- Beside defining functions ourselves, we can also call functions Python and other packages / 包 (e.g. NLTK) provide us with
- try:

```
text = 'He\'s a U.S. citizen. I\’m not.’
print(len(text))
```

  - what does the function `len()` do?
  - what if we want to find the number of words in above text? What about the number of sentences?

TEXT NORMALIZATION

TEXT NORMALIZATION

- Tokenization (especially for Chinese NLP tasks)
- Lemmatization & Stemming
- Sentence segmentation

TOKENIZATION

- try:

```
from nltk import word_tokenize
words = word_tokenize(text)
print(words)
print(len(words))
```

  Did it handle the 'U.S.' issue properly?

- NLTK: a package
  word_tokenize: a function in the NLTK package
- Let a user type in a sentence:

```
s = input("Enter some text: ")
print("You typed", len(word_tokenize(s)), "words.")
```

LEMMATIZATION & STEMMING

- What's the difference between the three?

```
import nltk, time
pt = nltk.PorterStemmer()
lcst = nltk.LancasterStemmer()
wnl = nltk.WordNetLemmatizer()
word = 'derivations'

t0 = time.time()
print(pt.stem(word))
t1 = time.time()
print(lcst.stem(word))
t2 = time.time()
print(wnl.lemmatize(word))
t3 = time.time()
print(t1-t0, t2-t1, t3-t2)
```

Complete the code to see which one works fastest!

SENTENCE SEGMENTATION

- try:

```
import pprint
text = nltk.corpus.gutenberg.raw('chesterton-
thursday.txt')
sents = nltk.sent_tokenize(text)
pprint.pprint(sents[80:90])
```

- What error did the tokenizer make?

PRACTICE

Try on your own, and we'll ask you about it next week!

- Complete the following function `quadratic(a, b, c)` that takes 3 parameters - a, b, c - and return the two roots of the equation $ax^2 + bx + c = 0$.
  Hint: use `math.sqrt()` to calculate square roots.

```
import math
def quadratic(a, b, c):
   ......
   return ......
```

THAT'S IT! CONGRATS!