PRACTICAL 1

PYTHON INTRO

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?

A SURVEY

- How many of you have…
  - coded in Python to complete a project?
  - coded in Python?
  - coded in at least one programming language?
  - not coded?

WHAT IS PROGRAMMING?

- 编程就象厨师烧菜，以红烧茄子为例：
- 原料（数据 / data）：茄子750克、肉片50克、葱姜蒜50克、酱油、盐、糖、鸡粉适量，水淀粉适量。
- 制作过程（算法 / algorithm）：
  - ✓ 将茄子切成块，葱姜蒜切末待用
  - ✓ 起油锅，待油热至八、九成时放入茄子，炸至茄子由硬变软时取出，将油沥干待用
  - ✓ 另起锅，锅中放油三汤匙，油热后，先爆香葱姜蒜，下肉片炒散，烹入酱油，加入少量水和鸡粉，放入茄子、盐、糖，大火煮开后改用小火煮至茄子入味，最后用淀粉勾芡
- 程序 ＝ 数据结构 ＋ 算法

cf. 谌卫军老师 <计算机程序设计基础>

WHY PYTHON?

- Power and control!
  - Various NLP tasks and resources
- Easy to learn
  - A (really) high-level programming language
  - Similar to natural language

NOW,
BE READY TO TYPE WITH ME

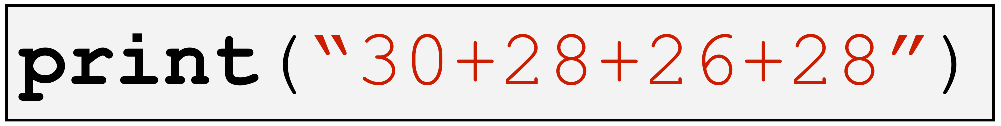
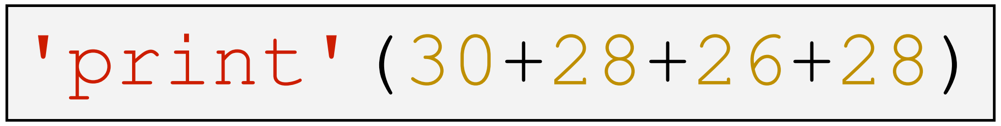
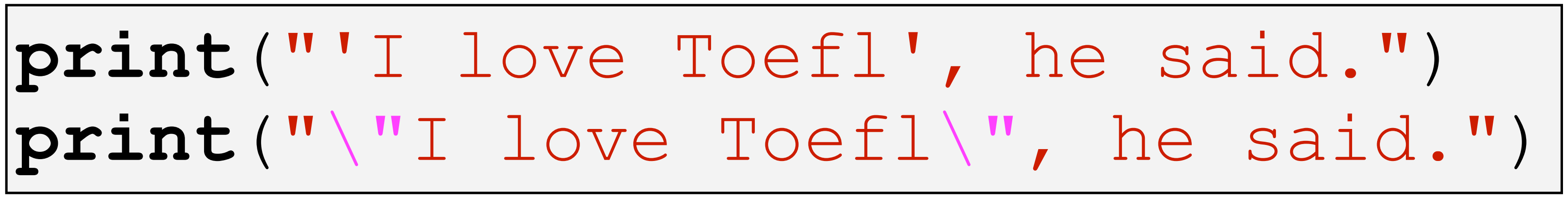
NUMBERS

- Integers (int) /整数 : a very self-explanatory name
  - e.g. 0, 1, -2896
- Floating Point (float) /浮点数: number with decimal points
  - e.g. 0.1, 1.0, -3.1415926
- Mathematical operations
  - try:

```
toefl_total = 30+28+26+28
print(toefl_total)
```

```
print(toefl_total==30+28+26+28)
```

STRINGS

- str/字符串: extremely superficial (only appearance matters)
  - e.g. “print”, “2.0”, “6+4”
  - try  `print(“30+28+26+28”)` and  `'print'(30+28+26+28)`
  - Don’t use Chinese quotes!!!
  - Either ‘’ or “” is fine, as long as no mixed usage
  - try 

```
print("'I love Toefl', he said.")
print("\"I love Toefl\", he said.")
```

  - what does \ do?

LISTS

- List / 列表 : a sequence (ordered) of items included in a pair of brackets: []
  - e.g. [], [1,2,3,4], [[1,2], 3, “4”]
  - Items in a list don’t have to be the same data type
  - try:

```
toefl_candidates = ['Jasmine', 'Jess', 'Maggie']
print(toefl_candidates[1])
```

Anything different than expected?

TUPLES

- Tuple / 元组: A sequence (ordered) of items included in a pair of parentheses: ()
    - very similar to lists, except that tuples are **immutable**
    - try:

```
toefl_secs = ('Reading', 'Listening', 'Speaking',
'Writing')
toefl_candidates.append('Ben')
print(toefl_candidates)
```

```
toefl_secs.append('Eating')
```

DICTIONARIES

- Dict / 字典：Items in lists/tuples are ordered (you can find an item by indexing), but in dictionaries, items are stored in the form of key-value pairs / 键-值对
  - e.g. {‘a’:1, ‘b’:2}
  - try

```
toefl_scores = {'R':30, 'L':28, 'S':26, `W':28}
```

```
toefl_scores['L'] += 2
print(toefl_scores)
```

A SUMMARY OF LISTS, TUPLES, AND DICTS

	LIST / 列表	TUPLE / 元组	DICT / 字典
Items	Chars, ints, floats, lists, tuples, dicts…	Chars, ints, floats, lists, tuples, dicts…	key-value pair
Ordered?	YES	YES	NO
e.g.	[30, 28, 26, 28]	(‘R’, ‘L’, ’S’, ‘W’)	{‘R’:30, ‘L’:28, ’S’:26, ‘W’:28}
IMMUTABLE?	NO	YES	NO
How to find an item?	Index	Index	Key

SEQUENCE OPERATIONS

- Strings, lists, and tuples are all sequence objects (sets of ordered items) in Python. On them, we can perform the following sequence operations:
  - Indexing
  - Slicing

INDEXING

- Each Item is indexed by their position (represented as integers).

```
print(toefl_candidates[0], toefl_candidates[-1])
```

- Again, ATTENTION: in Python we always count from 0! (You'll see its usefulness later)

SLICING

- A slice is a sub-sequence of a sequence object.
- Syntax for slicing: sequence[start:end:step]
  The start index is inclusive; the end index is exclusive. i.e. [start, end)
  - try

```
s = "abcdefg"
print(s[1:6:2])
```

- If the start/end index is not specified, its default value 0/-1 will be used. If the step index is not specified, its default value 1 will be used, and the colon before it can be omitted.
  - try

```
print(s[::2])
print(s[1:6])
```

PRACTICE

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

- Define `w1, w2, w3 = ‘I’, ’love’, ‘NLP’` .
    - Try to concatenate the words as a single sentence and store it in the string `my_sent` (hint: using the `+` operator).
    - Try `my_sent = ‘’.join([w1, w2, w3])`. Did it do the same thing? What problem has occured? How will you fix it?
    - `split()` works as an inverse method to `join()`. Can you use it to retrieve a `list` of the original words? Store the result in the variable `my_sent_2`.

PRACTICE

- Cont'd.:
  - Compare `my_sent` and `my_sent_2`. What’s their difference?
  - With `my_sent_2`, try to get the first 2 words in the sentence by indexing.
  - What does `my_sent_2[2][2]` do?
  - Which representation do you think is more useful in NLP tasks, `my_sent` or `my_sent_2?`

THAT’S IT!
CONGRATS!

WHAT'S MORE

- Beginning coding can be effortful, but YOU WILL get better soon. Before that, if you have any questions, you can
  - Search Stack Overflow: It's like 知乎 for coding. If you search for any question you have when learning this course, you'll find that a lot of people have already asked the same question in most cases!
  - Ask on Piazza: Plz don't use WeChat, cause it's much less powerful
  - Ask Google/Baidu/Bing…
- Other good Python learning resources:
  - 廖雪峰教程 / learnpython.org …