ENSF 592: Programming Fundamentals for Data Engineers

Yves Pauchard

Lecture 10: Goodies and Regex

October 7, 2019

Midterm

Oct 16, 2pm - 3:15pm. Location ENG 224

It will be a paper-based exam, 70min, 20pts. You will need a pen, and you are allowed a summary on a single page (letter, front and back). No calculators, phones, other devices.

Agenda

1. Key ideas in Ch 19
2. List and dict comprehension
3. Regex
4. Lambda functions
5. Preparation for next lecture

Notes: Key ideas Goodies

- Conditional expression `y = math.log(x) if x > 0 else float('nan')`
- List comprehension `[s for s in t if s.isupper()]` replaces

```
res = []
for s in t:
if s.isupper():
    res.append(s)
```

- Generator: create with `()` call `next(g)` to get the next element
- Generators passed to `sum()`, `min()`, etc.
- `any()` and `all()` take a sequence of boolean return one boolean.
- `set()` a collection of unique values
- `collections Counter` is a multi-set, or histogram
- `defaultdict` takes a factory to create new values
- `namedtuples` simple struct classes
- `*` gather positional `**` keyword arguments

Notes: Key ideas Regex

- Flexible pattern matching
- str.split() -> re.split()
- re.match() beginning of a string matches the pattern
- str.index() -> m = re.search(); m.start()
- str.replace() -> re.sub
- If you use backslashes in regex, use raw string r'\$'
- email = re.compile(r'\w+@\w+.[a-z]{3}')
- https://docs.python.org/3.7/howto/regex.html

List comprehension

Create new lists

```
[s for s in t if s.isupper()]
```

replaces

```
res = []
for s in t:
    if s.isupper():
        res.append(s)
```

Dict comprehension

Create new dictionaries

```
d = {0 : 'A', 1 : 'B', 2 : 'C', 3 : 'D'}
invert = {v : k for k, v in d.items()}
```

replaces

```
invert = {}
for k, v in d.items():
    invert[v] = k
```

Histogram with `set()`, generator and dict comprehension

```
x='otto'
y=set(x)
z={i:sum(j==i for j in x) for i in y}
print(z)
```

Regex

https://docs.python.org/3.7/howto/regex.html

```
^         Matches the beginning of a line
$         Matches the end of the line
.         Matches any character
\s        Matches whitespace
\S        Matches any non-whitespace character
*         Repeats a character zero or more times
*?        Repeats a character zero or more times (non-greedy)
+         Repeats a chracter one or more times
+?        Repeats a character one or more times (non-greedy)
[aeiou]   Matches a single character in the listed set
[^XYZ]    Matches a single character not in the listed set
[a-z0-9]  The set of characters can include a range
(         Indicates where string extraction is to start
)         Indicates where string extraction is to end
```

Regex

Write a code using RegEx that returns a subset of a given text that comes after the word “Python” in the text, but only if “Python” is not the very first word of the sentence and the sentence ends with full stop (“.”)

```
sample_text=["The Simplest Python Exam Ever.","Python 3.0.","Good Job Python!"]
for text in sample_text:
    print(re.findall('^.+Python (.*)\.$',text))
```

Uses `re.findall()` to get all non-overlapping occurrences.

Regex

Another example

```
text="Just another day in paradise."
X=text.split()
for w in X:
    if re.search('^.[^an].*',w):
        print(w)
# prints 'Just'
```

Match words with the second character not a or n

Uses `re.search()` to see if the word is a match.

Anonymous lambda functions

Create simple, one-off functions on the fly.

`lambda param: what-to-do`

used when passing functions as arguments

```
t = [2, 3, 1]
glued = list(zip(range(len(t)), t))
# [(0, 2), (1, 3), (2, 1)]
glued_sorted = sorted(glued, key=lambda x: x[1])
```

Next Lecture: Summary and Midterm prep