# Regular Expressions in Python

L555

Dept. of Linguistics, Indiana University
Fall 2010





## Regular Expression Module

### Module

In order to use regular expressions, we need to load the module.

```
import re
```





## Regular Expression Symbols

| | |
|---|---|
| . | wildcard |
| \ | escapes specials characters |
| [...] | character set |
| [^...] | complement of character set |
| \| | or |
| * | Kleene star |
| + | Kleene plus |
| { m,n } | repeat between m and n times |
| ^ | beginning of a string |
| $ | end of a string |





## Regular Expression Functions

| | |
|---|---|
| compile(<pattern>) | compiles a regex pattern into a pattern object – for reuse |
| search(<pattern>, <string>) | searches for regex pattern in string |
| match(<pattern>, <string>) | checks at **beginning** of string |
| split(<pattern>, <string>) | splits the string based on pattern, returns a **list** |
| findall(<pattern>, <string>) | returns a list of all occurrences |
| sub(<pat>, <rep>, <string>) | replaces pat by rep in string |





## Example

```
import re

mysent = raw_input('Gimme a sentence!')
if (not re.search('[ !\.;?]', mysent)):
    print 'this is not a sentence'
```





## Example

```
import re

mysent = raw_input('Gimme a sentence!')
newstr = re.sub('[A-Z]', 'XX', mysent)
print newstr
```

# Pattern Objects

## Module

The functions `compile`, `search`, and `match` return a pattern object. The objects contain information about the pattern itself and for the matching functions also information about the matched segments in the string.

```
import re

phoneNums = re.compile('^[(]?[0-9]{3,3}[-␣)]....
....[0-9]{3,3}[␣-][0-9]{4,4}$')

myphone = raw_input('Give␣me␣a␣phone␣number:␣')
if phoneNums.search(myphone):
    print 'format␣correct'
else:
    print 'format␣incorrect'
```





# Example

```
import re

mysent = '␣a␣rose␣is␣a␣rose␣is␣a␣rose'
allstr = re.search('(␣.␣)', mysent)
print allstr.group(1)
```