Regular expressions unit:

- Regular expressions—principles and Python (**today**)
- Lab: Building a RegEx-based chatbot (Wednesday)
- The edit distance algorithm [stand alone topic] (Friday)

Today:

- Wrap-up and review of concepts from last week
- Why we care about regular expressions
- Review and practice of regular expressions by definition
- Overview and demo of regular expressions in Python
**type.** A sequence of characters, independent of occurrence.

**token.** An occurrence of a type.

**lexeme.** A dictionary entry; a set of types associated together with a definition, etymology, etc.

**wordform.** One of the associated types of a lexeme; an inflectional form of a lexeme.

**lemma.** The headword in a dictionary entry; a wordform that serves as the canonical representative of a lexeme.

**corpus.** A collection of texts; a dataset for natural language processing.

**vocabulary.** The set of types in a corpus.

J&M ambiguously uses *lemma* to mean either lemma or lexeme.

*Word type or word token* are sometimes used to distinguish from other uses of the terms *type* and *token.*
- An **alphabet** is a set of symbols, $\Sigma$.
- A **string** over an alphabet is a sequence of symbols from that alphabet. $\Sigma^*$ is the set of all strings over alphabet $\Sigma$.
- A **language** over an alphabet is a set of strings, that is, a subset of $\Sigma^*$.

- **Regular expressions** constitute a system for specifying languages. (J&M, “a language for specifying text search strings”, pg 3.). An individual regular expression denotes a language, that is, a set of strings.
base cases

- $\emptyset$ the empty set of strings
- $\varepsilon$ the set containing the empty string, $\{""\}$
- $a$ the set containing only the string with only $a$, for some $a \in \Sigma$, $\{"a"\}$

recursive cases

- $rs$ the set of strings made from concatenating strings from $r$ and $s$, $\{x + y \mid x \in r \land y \in s\}$, for some regular expressions $r$ and $s$
- $r|s$ the set of strings from $r$ or $s$, $r \cup s$
  for some regular expressions $r$ and $s$
- $r*$ the set of strings made from concatenating 0 or more strings from $r$
  for some regular expression $r$
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>[abc]</td>
<td>One occurrence of any of these symbols</td>
<td>(a</td>
</tr>
<tr>
<td>[a–c]</td>
<td>One occurrence of any symbol in this range</td>
<td>(a</td>
</tr>
<tr>
<td>r?</td>
<td>Optionally an occurrence of a string defined by r</td>
<td>(r</td>
</tr>
<tr>
<td>r^5</td>
<td>5 occurrences of a string defined by r</td>
<td>rrrrr</td>
</tr>
<tr>
<td>r^{3,5}</td>
<td>Between 3 and 5 occurrences of a string defined by r</td>
<td>(rrr</td>
</tr>
<tr>
<td>r+</td>
<td>One or more occurrences of a string defined by r</td>
<td>rr*</td>
</tr>
</tbody>
</table>
DNA sequences: \((A|C|G|T)\)^* 

Identifiers: \((\ ' \mid \varepsilon \mid [A-Za-z][A-Za-z0-9_]\)\)\_ 

Phone numbers: \([2-9][0-9]^2 - [2-9][0-9]^2 - [0-9]^4\) 

Dates: \((1[0-2]) | [1-9]/(30|31|([12][0-9]) | [1-9])/[1-9] [0-9]^{0,3}\) 

US Postal Addresses: \([0-9]+ [NSEW]^{0,2} [A-Z][a-z]* (St|Ave|Rd|Ln|Dr|Blvd), ([A-Z][a-z]*)*, [A-Z]^2 [0-9]^5\)
Lord, you have been our dwelling place in all generations.
Coming up:

- Read J&M, Sections 2.(0–4) (Mon, Aug 28)
- Python warm-up assignment (Tues, Aug 29)
- Regular expressions quiz (Tues, Aug 29)
- Words and corpora quiz (Thurs, Aug 31)
- Read J&M, Section 2.5 (Fri, Sept 1)
- Regular expressions assignment (Fri, Sept 1)

Next time: Regular expression chatbot in the lab.