This quick reference gives a concise overview of the most commonly needed features of **Simple Query Syntax**. Query expressions that you can enter in CQPweb’s search box are printed in typewriter font, followed by an arrow and the matching words or word sequences in italics (e.g. `st?ing` ➔ `sting, stung`).

### Basic word form searches

- To search for word forms, simply type them into the query field and click [Start query]: `glitterati` ➔ `glitterati`  
- Use wildcards for unspecified letters, and prefix or suffix searches:

  - `?` for a single arbitrary character  
    - `s?ng` ➔ `sing, sang, song, …`  
  - `*` for zero or more characters  
    - `*able` ➔ `able, table, capable, suitable, available, …`  
  - `+` for one or more characters  
    - `+able` ➔ `table, capable, suitable, … but not able`  
  - `??+` for three or more characters, etc.  
    - `??+able` ➔ `capable, … but not able, table, unable, stable`  

- Combine multiple wildcards: `*oo+oo*` ➔ `Voodoo, schoolroom, …`  
- Protect wildcards and other metacharacters with backslash \ to match the literal character (called "escaping" the metacharacter):
  
  - `\?` ➔ `?`  
  - `?` ➔ `a, b, c, …, A, B, C, …, 1, 2, 3, …, !, ?, …`  

Simple Query Syntax uses the following metacharacters:

- List comma-separated alternatives (optionally including wildcards) in square brackets:
  
  - `??+[able,ability]` ➔ `capable, capability, availability, …`  
  - `neighbo[u,r]r` ➔ `neighbour, neighbor`  

- Searches are case-insensitive by default: the queries `bath`, `Bath` and `BATH` find the same matches (i.e. all instances of the three word forms `bath`, `Bath` and `BATH`). Set the “Query mode” drop-down to “Simple query (case-sensitive)” to distinguish `AIDS` and `aids`, for example.  

- Use `:c` modifier to ignore case on just part of a case-sensitive query:
  
  - `The bath:c` ➔ `The bath, The Bath, The BATH`  

- Use `:d` modifier to ignore accents: `fiancée:d` ➔ `fiancée, fiancee`
Matching parts-of-speech (POS)

- Search for a word form with a specific POS tag by linking them with an underscore _. Wildcards can be used both for word form and POS tag:

  \[ \text{lights}_\text{NN2} \rightarrow \text{plural noun lights, but not the verb form lights} \]
  \[ *\text{ly}_\text{RR} \rightarrow \text{adjectives ending in -ly (e.g. daily)} \]
  \[ \text{super+}_\text{V*} \rightarrow \text{verb forms starting with super-} \]

- You can also search by POS tag only: \_\text{NN1} \rightarrow \text{any singular noun}

- **Warnings**: (1) Different corpora use different tagsets; the examples here use the C6 tagset. (2) Some corpora may not be tagged at all. (3) Some corpora may use the _ symbol for a different kind of tag (not POS but something else)

- You will find links to descriptions of the tagsets in use in a particular corpus in the main menu under the heading “Corpus Info”.

- Some commonly-used POS tagsets are listed at the end of this document.

- Keep in mind that part-of-speech tags are likely to have been assigned by an automatic software tool and are not always correct (try e.g. can\_\text{NN1}).

Matching simplified POS tags

- Use simplified POS tags enclosed in curly braces: super+_\{VERB\} for verb forms starting with super- (no wildcards allowed in simplified tags).

- List of simplified POS tags:

  \[
  \begin{array}{llllll}
    \text{A, ADJ} & \text{adjective} & \text{INT, INTERJ} & \text{interjection} \\
    \text{N, SUBST} & \text{noun} & \text{PREP} & \text{preposition} \\
    \text{V, VERB} & \text{verb} & \text{PRON} & \text{pronoun} \\
    \text{ADV} & \text{adverb} & $, \text{STOP} & \text{punctuation} \\
    \text{ART} & \text{article} & \text{UNC} & \text{other / uncertain} \\
    \text{CONJ} & \text{conjunction} \\
  \end{array}
  \]

- **Warnings**: as with normal tags, the tagset may vary (the simple tags above are the Oxford Simplified Tagset), or not be available at all, or the _{} symbol may be used for a different kind of tag

- Simplified POS tags are prone to the same errors as normal POS tags.
**Lemma queries**

- Search by lemma (i.e. dictionary headword), enclosed in curly braces: `{light}` finds the forms `light`, `lights`, `lit`, `lighted`, `lighting`, `lighter` and `lightest` (but not the nouns `lighting` and `lighter`).

- The lemmatization scheme may vary depending on the language and the corpus. Look at the frequency list for lemma if in doubt.

- You can combine lemma and simple tag queries using a slash:

  - `{light/V}` ➔ `light`, `lights`, `lit`, `lighted`, `lighting` (tagged as verb)
  - `{light/N}` ➔ `light`, `lights` (tagged as noun)
  - `{light/A}` ➔ `light`, `lighter`, `lightest` (tagged as adjective)

- **Warning**: in some corpora, the `{ }` may be used for an annotation other than lemma.

- Lemma errors can arise in the same way as POS tag errors.

**Word sequences**

- Queries can consist of multiple words, e.g. `talk of the town`

- All words and punctuation symbols (“tokens”) are separated by blanks; possessives (`Peter's`) and contracted forms (`they've`, `gonna`) are usually split (in most corpora):

  ```
  he will \, wo n't he \? ➔ he will, won't he?
  ```

- Each query item in a sequence can make full use of wildcards, part-of-speech constraints, and headword or lemma searches:

  ```
  {number/N} of _{A} _NN2 ➔ numbers of younger men, ...
  ```

- Use + to skip an arbitrary token, or * for an optional token. Combine + and * for larger gaps, e.g. +++** to skip between 3 and 5 tokens.

  ```
  {eat} * up ➔ eat up, ate up, eat it up, eaten all up, ...
  {eat} + up ➔ eat it up, eaten all up, ... but not eat up, ate up
  {eat} +++ up ➔ up at a distance of 3 or 4 tokens after eat
  ```

- Use ! to negate a query item, specifying any token that doesn’t match it:

  ```
  {go} !mad ➔ go along, goes home, ... but not going mad
  {!go} mad ➔ be mad, some mad, ... but not gone mad
  leaves_!VVZ ➔ leaves as a noun but not as a verb
  !on_{PREP} fire ➔ in fire, from fire, ... but not on fire
  ```
Advanced lexico-grammatical patterns

- Use regular expression notation for alternatives, optional elements and repetition within a sequence:

Examples using simplified POS tags (see above):

\[
\begin{align*}
(_{A})? & \quad \text{optional adjective} \\
(_{A})^{*} & \quad \text{zero or more adjectives (optional)} \\
(_{A})^{+} & \quad \text{one or more adjectives (non-optional)} \\
(_{A})^{2,4} & \quad \text{between two and four adjectives} \\
(...|...|...) & \quad \text{matches one of the alternatives indicated by } ... \\
(...|...|...)^{*} & \quad \text{alternatives with repetition (optional)} \\
(...|...|...)^{+} & \quad \text{alternatives with repetition (non-optional)} \\
(...|...|...)^{2,4} & \quad \text{between two and four repetitions of the given alternatives (may be in any order)}
\end{align*}
\]

- Regular expression notation can be nested to match complex patterns:

\[
\text{the (most } _{AJ0} | _{AJS} \{\text{man}}
\rightarrow \text{the biggest man, the most attractive man, ...}
\text{the (most } (_{AV0})? _{AJ0} | (_{AV0})? _{AJS} \{\text{man}}
\rightarrow \text{plus: the very richest men, the most supremely stupid men, ...}
\]

- Complex syntactic patterns can be formed, e.g. for a prepositional phrase:

\[
_{\text{PREP}} (_{\text{ART}})? ((_{\text{ADV}})? _{\text{A}})^{*} _{\text{N}}
\]

"a preposition; followed by an optional article; followed by any number of adjectives (zero or more), each of which may optionally be preceded by an adverb; followed by a noun"
XML tags

- XML start and end tags can be inserted in query expression to match the boundaries of a region, e.g. the start \(<s>\) or end \(</s>\) of a sentence:

  \(<s>\) but \(\Rightarrow\) sentence beginning with \(\textit{but}\) (or \(\textit{But}\))

  \(_{($)} <s>\) \(\Rightarrow\) punctuation mark at end of sentence

- To match a complete region, skip all tokens between the start and end tag:

  \(<\text{quote}> (+)+ </\text{quote}> \Rightarrow \text{list of all quotations}\)

  \(<\text{mw}> (+)+ </\text{mw}> \Rightarrow \text{list of all multiword units}\)

- Some commonly-used XML tags:

  \(<s> \ldots </s> \text{sentence}\)

  \(<p> \ldots </p> \text{paragraph}\)

  \(<u> \ldots </u> \text{speaker turn}\)

  \(<\text{head}> \ldots </\text{head}> \text{heading or caption}\)

- You may be able to find documentation regarding the XML available in a particular corpus in the links under “Corpus Info” on the main menu

Proximity queries

- Special syntax for searching one item within a specified range of another:

  \(\textit{kick} <s> \textit{bucket} \Rightarrow \textit{kick} \text{ and } \textit{bucket} \text{ in the same sentence}\)

  \(\{\textit{kick/V}\} <s> \textit{bucket}\_NN1 \text{ (can use POS/lemma constraints)}\)

  \(\textit{day} <5<< \textit{night} \Rightarrow \textit{day} \text{ and } \textit{night} \text{ within range of 3 tokens}\)

  \(\textit{day} >>5>> \textit{night} \Rightarrow \textit{day} \ldots \textit{night} \text{ (within 5 tokens)}\)

- Only the left element (“target”) will be highlighted on the result page. The right element is considered as a “constraint” that must be satisfied.

- Multiple constraints can be chained:

  \(\{\textit{day}\} <5>> \{\textit{month}\} <5>> \{\textit{year}\}\)

  In this case, \textit{day} must co-occur with \textit{month} as well as \textit{year} in a 5-token window; only \textit{day} will be highlighted in the concordance.

- Proximity queries can be nested with parentheses:

  \(\{\textit{waste/V}\} <s> (\textit{time} <3>> \textit{money})\)

  Here, the verb \textit{waste} must co-occur with \textit{time} as well as \textit{money} in the same sentence; but \textit{time} and \textit{money} must be closer together (within a 3-token window). Again, only instances of \textit{waste} will be highlighted.

- Proximity queries cannot be combined with lexico-grammatical patterns!
Some commonly used part-of-speech tagsets

The following list of tagsets is very far from comprehensive, but contains a few tagsets commonly used for some major languages:

**English**
- [C6 tagset](#) (normal tagset used by the CLAWS tagger) (related: C7)
  - Used for all the examples in this document
- [C5 tagset](#) (used for the BNC; has fewer tags than C6)
- [C8 tagset](#) (more fine-grained version of C6)
- [Brown Corpus tagset](#) (early and influential)
- [Penn Treebank tagset](#) (see also) (similar to the four above, but simpler; used by Treetagger for English)
- [ICE tagset](#) (designed as basis for syntactic analysis)

**Chinese**
- [LCMC tagset](#) (used by TreeTagger for Chinese)
- [Penn Chinese Treebank tags](#)

**Arabic**
- [Lancaster system for Arabic corpora](#) (based on MADA tagger output)
- [Buckwalter analysis](#) (see also)

**Russian**
- [MULTEXT-East tags](#) for Russian

**German**
- [STTS tagset](#)
- [TIGER TAGSET](#)

**Italian**
- [TreeTagger](#) tagset for Italian
- [DMI codes](#)

But wherever possible, you should look for the links on the left-hand-side menu on the begin-query screen in CQPweb, as these links should be tuned to the set-up of the specific corpus!