# Advanced Search Operators

(Oct 5, 2021)
Daniel M. Russell

Here, in one place, are all of the currently documented advanced search operators. *Note that some operators come in two versions of the same operator:* (e.g., **allinanchor** along with **inanchor:** ). I've written about them together rather than having two entries for the same kind of operator. (But, in truth, I almost never use the "allin…" form.)

Note: In this document, I follow the square brackets convention, here a query is surrounded by square brackets. So, when doing the query, you wouldn't actually use the square brackets in your query. (Although it won't hurt anything either…) Example:

[ codfish site:nytimes.com ]

--------------------------------------------------------------------

**allinanchor: / inanchor:**
-- Google restricts results to pages containing all query terms in the anchor text on links to the page. For instance: [ **allinanchor**: best restaurant Sunnyvale ] will return only pages in which the anchor text on links to the pages contain the words "best" "restaurant" and "Sunnyvale" – that is, all of the words following the **allinanchor** operator. So, when using **allinanchor**: in your query, do not include any other search operators. By contrast, using the operator **inanchor**: only searches for the term that's next. (This is true for all of the "all…" operators.) Example: [ **inanchor**:sales offer 2011 ] will search only for "sales" in the anchor text.

> Definition: Anchor text is the text you see on a page that is linked to another web page or a different place on the current page. For instance, this [sample link](#) has "sample link" as its anchor text. When you click on anchor text, you will be taken to the page or place on the page to which it is linked.
>
> Example: [ **allinachor: best restaurant La Jolla ocean view** ] - search for those terms in the anchor

**allintext: / intext:**
-- restricts results to those containing all the query terms you specify in the text of the page. For example, [ **allintext**: camping tent stove] will return only pages in which the words "camping" "tent" and "stove" appear in the text of the page. Using the operator "**intext:** will search only for the next term in the text of the page. . (Note: using **intext**: in front of every word in your query is the same as using allintext: at the front of your query, e.g., [ **intext**:Victorian **intext**:artists ] is the same as [ **allintext**: Victorian artists ].)

Example:  **[ allintext:mesothelioma asbestos symptoms ]** - search for all those terms on the page

**allintitle: / intitle:**

-- results to those containing all the query terms you specify in the title. For example, [ **allintitle**: university relations ] will return only documents that contain the words "university" and "relations" in the title of the page.  Using the operator **intitle:** will search only for the next term in the title of the page.  For instance, [ flu shot **intitle**:help ] will return documents that mention the word "help" in their titles, and mention the words "flu" and "shot" anywhere in the document (title or not).

**allinurl: / inurl:**

-- restricts results to those containing all the query terms you specify in the URL. For example, [ **allinurl**:google faq ] will return only documents that contain the words "google" and "faq" in the URL, such as "[www.google.com/help/faq.html](www.google.com/help/faq.html)".

*term1*  **AROUND**  *n*  *term2*

-- limits results to those documents where *term1* appears within a certain number of words of *term2*.  For instance,

[ search **AROUND** 3 engine ]

will find only documents that have the words "search" within 3 words of "engine" – this is particularly useful when searching for common words that are relevant to your search only when in close proximity. (For instance, useful for finding two names in close quarters:  [ "Politician Surname" AROUND 5 "Major Donor" ] )  Note that this doesn't preserve order: it will find "engine search" and "search engine"

**before:**

-- lets you find results that were published before a given date.  Example:

[ avengers endgame **before:**2018-1-1 ]

where the date is YYYY-MM-DD - note that if you only specify the year, it will default to the first of that year.  And likewise…

**after:**

-- lets you find results that were published after a given date.  Example:

[ avengers endgame **after:**2020-1-1 ]

where the date is YYYY-MM-DD - note that if you only specify the year, it will default to the first of that year.

**define**
-- gives definitions from pages on the web for the term that follows. Useful  for finding definitions of words, phrases, and acronyms. For example, [ **define** peruse ] will  give a definition of the word "peruse."  This also works for many phrases,   (Note:  This operator used to be "define:" -- we got rid of the colon a while ago.)

[ **define** Hobson's choice ]

Of course, if you just search for a rare word (for example, [ surfeit ]) you'll get the definition as the first search result.

**filetype:***suffix*
  -- limits results to pages whose names end in **suffix**.  The suffix is anything following the last period in the file name of the web page and can be many characters in length.

Example:  [ search engine guidelines **filetype**:pdf ] will return Adobe Acrobat pdf files that match the terms "search," "engine," "guidelines," and are  pages whose names end with pdf. Note that the filetype: operator works with any string after the last period, thus this works: [filetype:1234567] Not useful too often, unless you know the final string of a document.

[ **filetype:pptx** superconductor lesson plan ]

Interesting note:  You can't find MP3 files this way any longer.  It used to work, but then copyright issues started raising their heads.  The closest you can get to this now is to search the URL like this:  [ inurl:mp3 ]

***Fill in the blanks* (*** - the "star operator")**
-- the *****, or wildcard, is a little-known feature that can be quite useful. If you include ***** within a query, it tells Google to try to treat the star as a placeholder for any unknown term(s) and then find the best matches, up to 5 terms in length. For

Example: **[ Google * ]** will give you results about many of Google's products (go to next page and next page -- we have many products). The query **[ Obama voted * on the * bill** ] will give you stories about different votes on different bills. Note that the ***** operator works only on whole words, not parts of words.  (See one of the Search Idioms below for an advanced use of ***** in a quoted query, such as **[ "Google * products"** ] .)

**inanchor:**   (see **allinanchor:** above )

**intext:** (see **allintext:** above)

**intitle:** ( see **allintitle:** above)

**inurl:** (see **allinurl:** above)

**Minus sign**  ( **–** ) to exclude
        -- placing  a minus sign immediately before a word indicates that you do not want pages that contain this word to appear in your results. The minus sign should appear immediately before the word and should be preceded with a space. For example, in the query [ anti-virus software ], the minus sign is used as a hyphen and will not be interpreted as an exclusion symbol; whereas the query:  [ anti-virus –software ] will search for the words 'anti-virus' but exclude references to software. You can exclude as many words as you want by using the – sign in front of all of them, for example [ jaguar –cars –football –os ]. The – sign can be used to exclude more than just words. For example, place a minus sign before the 'site:' operator (without a space) to exclude a specific site from your search results.

**Number range** ( .. )
 --  The number range operator searches for results containing numbers in a given range. Just add two numbers, separated by two periods, with no spaces, into the search box along with your search terms. Example: [  Willie Mays 1950..1960 ]   You can also specify a unit of measurement or some other indicator of what the number range represents.  For example, here's how you'd search for a DVD player that costs between $50 and $100:

    [ DVD player $50..$100 ]

**OR**
        -- the Boolean operator OR specifies alternatives to use as synonyms in search.  For instance, the query:

    [ mesothelioma **OR** "lung disease"  treatment ]

could be used to search for a treatment for either mesothelioma or the quoted phrase "lung disease" (Be sure to make the OR all uppercase.  Lowercase *or* doesn't work.)  Note that OR works *inside of a quoted phrase.*  That is, a query like  [ "jackfruit OR papaya" ] is not a phrase search… it's the same as [ jackfruit OR papaya ]

*Phrase search* (using double quotes, "…" )
  -- by putting double quotes around a set of words, you are telling Google to consider the exact words in that exact order without any change. Google already uses the order and the

fact that the words are together as a very strong signal and will stray from it only for a good reason, so quotes are usually unnecessary. By insisting on phrase search you might be missing good results accidentally. For example, a search for [ "Alexander Bell" ] (with quotes) will miss any pages that refer to Alexander G. Bell.

Try this to see how it works:
    [ "Alexander * Bell" ]
        -- this will give you results like "Alexander Graham Bell" and "Alexander G. Bell" and
           "Alexander le Bell"


***Search exactly as is (double quotes around a single word) -*** **aka 'phrase search'**
        -- Google employs synonyms automatically, so that it finds pages that mention, for example, childcare for the query [ child care ] (with a space), or California history for the query [ ca history ]. But sometimes Google helps out a little too much and gives you a synonym when you don't really want it. By attaching double quotes around a word as in the query

    [ "ca" history ]

you are telling Google to match that word precisely as you typed it.  Note that you can put double quotes inside of double quotes, as in:

    [ "words that are "mispelled" by" ]

this query searches for the phrase "words that are mispelled by" -- but the term "mispelled" is **intentionally** misspelled.  (Get it?)


**site:**
        -- using the **site:** operator restricts your search results to the site or domain you specify. For example, [ penquins site:.aq ] will search for pages about penguins from web sites that have an AQ top-level domain name.  (AQ is Antarctica, and is mostly research stations located there.) A query like  [ accidents **site**:bls.gov ] will find pages about accidents within the bls.gov domain (BLS = Bureau of Labor Statistics).  You can specify a domain with or without a period, e.g., either as .gov or gov.

**related:**
        -- a search for  **related**:URL lists pages that are similar to the web page you specify. For instance, [ **related**:en.wikipedia.org] will list web pages that are similar to the Wikipedia homepage.  (**Caution**:  This is in the process of being deprecated.  August, 2019)

***Combinations of operators:***
    -- many of the search operators  –, OR, and " " can be combined.  For example, to find articles on security from all sites except Wikipedia.org you would search for:
[ article security –site:Wikipedia.org  ]  Similarly, you might want to exclude some kinds of documents with a search such as [ salsa recipe -tomatoes -filetype:pdf ] which would find salsa recipes that do not include the term "tomatoes" and are not PDF files.

***More advanced search options*:**
        -- Note that the Advanced Search page ([http://www.google.com/advanced_search](http://www.google.com/advanced_search)) also provides a set of search options that are not available as special operators.  Using the Advanced Search page you can also:

        - filter by language (e.g., find pages only in Spanish, Chinese, German, etc.)
        - date (filter by time)
        - usage rights (filter by Creative Commons license)
        - reading level (find pages that are Basic, Intermediate or Advanced reading levels)

**Handy search idioms:**
        -- a "search idiom" is a particular way of doing a search that's sometimes really handy.
Idiom 1:  ***Site minus site.***  A search like:  [ site:nyc.gov -site:www.nyc.gov ] will give you sites in NYC.gov that do NOT begin with WWW.  That's handy for finding subdomains within a particular site, which can you *then* use **site:** to search.

Idiom 2:  ***Stars in quoted phrase***.  A query like [ whenever * says * whenever ] looks for words in roughly that sequence.  But if you *really* want words in that order (and this is what I usually do), you want to use the stars *within* a quoted phrase.

Example:  [  "whenever * says * whenever" ] is a very different query.  Look at them side-by-side.  The one on the right is a much better query, for my money.
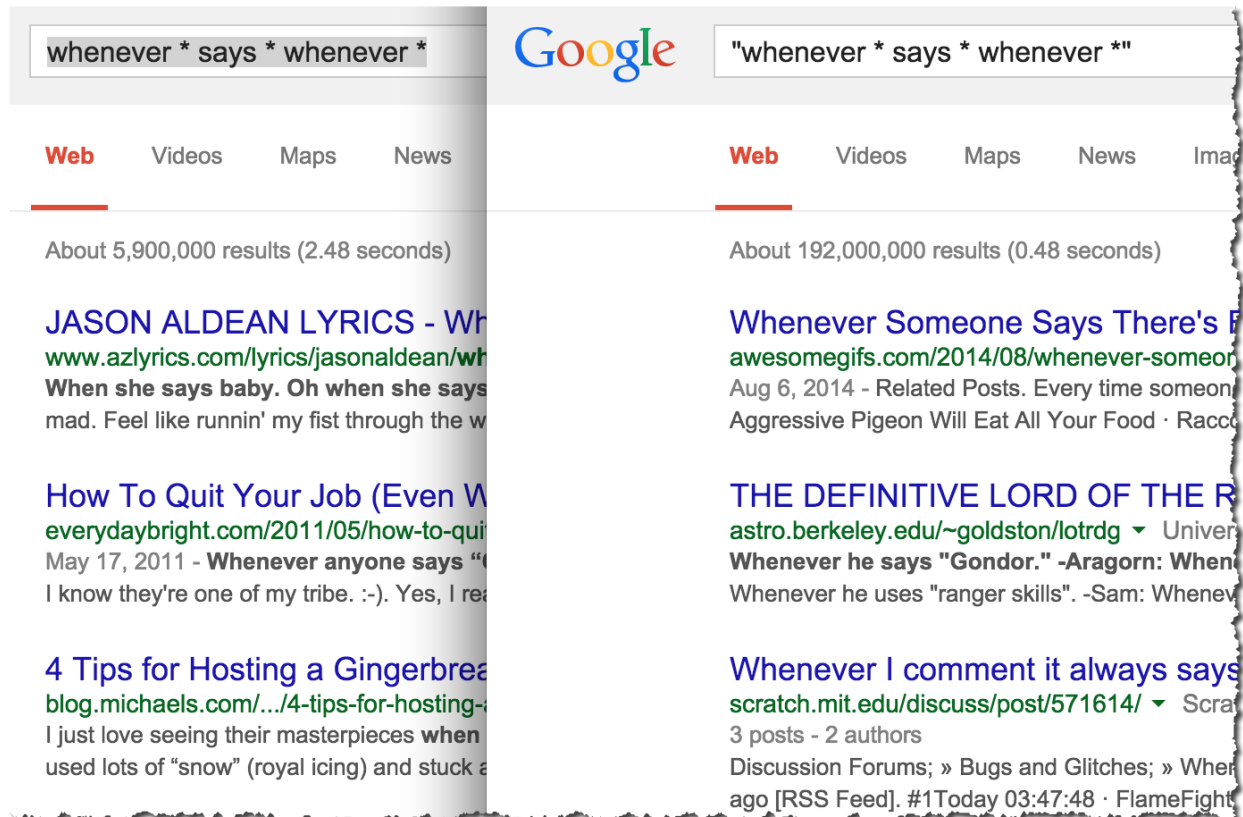
Idiom 3: **Stars in site search**.

A search like [ site:*.law.*.edu ] will find all of the .EDU sites with ".law" in the domain name.

Also try:  [ site:*.nyc.gov ] will match all of the NYC.gov sites with a subdomain.

Also:  [ site:*.nasa.*  inurl:education ] gives lots of good clues about education sites at NASA.

Generally, a * matches any token (that is, an entire term without spaces).  A token would be something like: *www  nasa  gov  schools  spartan-resume*

So you can always replace a token in a URL by a *, and it will match ANY token that can/would fit into that position.

But, oddly enough, * *also matches* prefix strings before tokens.. except for the TLD (the last term in a URL).

Thus, queries like this will work:
    [ site:www.*creative.com ] -- will match sites like
        **www.av-creative.com**

**www.aquila-creative.com**

While queries like:
  [ site:www.*creative.* ] -- will also work to show sites like
    **www.dei-creative.biz**
    **www.afr-creative.com**


Oddly, the * doesn't seem to match anything in the *post* position (e.g., [ site:foo*.com ] ).  I can imagine why that's true (it's easier to match if it's prefix *).


**NOTES:**

If you've read this far, you're a true Search Nerd.  Nicely done.

***Special characters:***  Not long ago, only a few very special characters were searchable.  These days, almost all of them are searchable!  You can now search for ∞   Note that a few symbols: $ %  # are special-special and can be searched for as part of a term (e.g.,   #cute or $50 or 24%).  See "Number Range" above.

Another special-special character is : (colon).  You CAN search for things that look like this: [ 10:27 ] (useful mostly for Bible verses).

But generally speaking, you can search for extended characters.
Examples:
  [ I ❤  NY ]
  [ 🍺 pub ]  or, the logical extension of this:   [ 🍺 nagano site:.jp ]
  [ size  7½ ]

(But note that the ½ character will be synonymized to 0.5 as well, so you might find 7.5 as well as 7½ -- if you don't want the synonyms, use Verbatim mode OR the double quote operator).

A *special-special* character is + -- you can now search for special terms that we have with a plus character in it in the last position.  Examples:

  [ G+ ]
  [ C++ ]
  [ Coke+ ]
  [ timer+ ]
  [ 40+ ]   -- and other numbers (20+,30+, 55+, etc.)

***Parentheses don't matter:*** I know lots of people want to be able to do something like this:

    [ (A B ) OR (C D) ]

but Google doesn't see the parens, and that gets turned into:

    [ A B OR C D ]

which means just A, (B OR C), and D. If you want to get [ A B ] OR [ C D ] you'll have to do 2 separate searches and combine the results yourself, which is probably what you really want to do. Why doesn't Google support this? Because we've found that *most people get it wrong*. That is, when they use parens, they end up with a search that is much worse than if Google ignores the parens. (I've seen this in my own field studies, even among information professionals--people mess up the parens a LOT.)

***Word order:*** Yes, word order matters in a search. Consider the difference between the searches: [ to be or not to be ] vs. [ be to not or be to ] -- the results are VERY different.

-----
## Deprecated:

**Link:**  (taken out of service mid-2016)
          -- **link**:URL shows pages that point to that URL.
          -- For example, to find pages that point to Google's 3DWarehouse home page you used to  search for…
             [ **link:** http://sketchup.google.com/3dwarehouse/ ]

        But it was getting really abused, so Google had to remove it from general use.


**Info:**  removed in mid-2017


**Plus (+) operator:** And you're probably wondering what happened to the + operator.  Well, here's the story.

It USED TO MEAN the same as Verbatim (i.e., do not spell correct, and use no synonyms).  Then, during the age of Google-+ (G+) it used to be a flag for "Search G+ for mentions of this term."  But that interpretation is long gone.


**Synonym (~) tilde operator:** This used to let searchers ask for "and all the synonyms" of a term.  (example:  [ ~mesothelioma ] would search for **mesothelioma** and all of its synonyms.  That was removed because synonyms is always turned on these days.  It would just be redundant.