

Quarto Postcodes (with inclusion)

```
import json
from letters import nL
from tdda.utils import dict_to_tex_macros

RE = r'^[A-Z]{1,2}[0-9]{1,2}[A-Z]? [0-9][A-Z]{2}$'

def n_poss_postcodes_for_re():
    """
    Number of strings matching:
    ^[A-Z]{1,2}[0-9]{1,2}[A-Z]? [0-9][A-Z]{2}$
    """
    n_postal_areas = nL + nL * nL # 1 or two letters
    n_postal_districts = 10 + 100 # Any one or two digit number
    # 0 and 0x aren't used, but match the regex
    n_subdistricts = nL + 1 # Not all letters are used,
    # and only for some London codes,
    # but for our regex...
    # The +1 is for ones not using a subdistrict

    n_outcodes = n_postal_areas * n_postal_districts * n_subdistricts
    n_incodes = 10 * nL * nL # Digit then two letters
    n_postcodes = n_outcodes * n_incodes

    return n_postcodes

if __name__ == '__main__':
    n = n_poss_postcodes_for_re()
    d = {'n': n, 'n_str': f'{n:,}', 'postcodeRE': RE}
    json_path = 'postcodes.json'
    with open(json_path, 'w') as f:
```

```
    json.dump(d, f, indent=4)
    dict_to_tex_macros(d, 'postcodes-defs.tex', verbose=False)
```

```
with open('ref/python_postcodes_py/postcodes.json') as f:
    ref = json.load(f)
assert d == ref
```

The number of postcode-like strings matching

$$^{\wedge}[A-Z]\{1,2\}[0-9]\{1,2\}[A-Z]?[0-9][A-Z]\{2\}\$$$

is 14,094,194,400