Regular Expressions

A regular expression or regex is a sequence of characters that define a search pattern. Usually this pattern is then used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation.

Regular expressions are used in search engines, search and replace dialogs of word processors and text editors, in text processing utilities such as AWK and in lexical analysis. Many programming languages provide regex capabilities, built-in or via libraries.

Python has very strong support for regular expressions.

d - digit [0 -9]
w - alphanumeric [a-zA-Z0-9_]
* - 0 or more
+ - 1 or more
? - 0 or 1

Rules of game:

1. **Search happens from left to right.**
2. **Search has 2 criteria - minimum criteria and maximum criteria which can both be same when you look for exact pattern. The moment minimum criteria is satisfied search becomes successful. Of course while returning it would try to look for maximum criteria. (greedy approach)**

- **\d+ means 1 or more digits. so minimum criteria is 1 digit and maximum criteria is infinite digits. So the moment first digit is encountered search becomes successful.**
- **\d* means 0 or more digits. so minimum criteria is nothing which means search always becomes successful at beginning of string irrespective of anything. here maximum criteria is infinite digits.**
- **if u look for exact string as pattern like “python”, minimum and maximum criteria are both same.**

Example:

```
import re
>> str1 = "my mobile number is 9876543210"
>>> b = re.search("\d+",str1)
>>> b.start()
20
>>> b.end()
30
>>> b.group()
'9876543210'
>>> b = re.search("\d*",str1)
>>> b.start()
0
>>> b.end()
0
>>> b.group()
''
>>> str1 = "9876543210 is my mobile number"
>>> b = re.search("\d+",str1)
>>> b.group()
'9876543210'
>>> b = re.search("\d*",str1)
>>> b.start()
0
>>> b.end()
10
>>> b.group()
'9876543210'
```