Python Regex Cheat Sheet

1. Basic Matching

. Any character except newline Example: a.c -> abc, axc
^ Start of string Example: ^Hi -> matches "Hi there"
$ End of string Example: end$ -> matches "The end"
[] Any one character inside Example: [aeiou] -> vowels
[^] Not any character inside Example: [^0-9] -> non-digits
\ Escape special character Example: \. -> literal dot

2. Predefined Character Classes

\d Digit (0-9)
\D Non-digit
\w Word char (letters, digits, _)
\W Non-word char
\s Whitespace
\S Non-whitespace

3. Quantifiers

* 0 or more Example: a* -> "", a, aaa
+ 1 or more Example: a+ -> a, aa
? 0 or 1 Example: a? -> "", a
{n} Exactly n Example: \d{4} -> 2025
{n,} n or more Example: \d{2,} -> 42, 1234
{n,m} Between n and m Example: \d{2,4} -> 42, 1234

4. Groups & Alternation

() Group for extraction Example: (ab)+ -> abab
| OR operator Example: cat|dog
(?:) Non-capturing group Example: (?:abc){2} -> abcabc

5. Common Patterns

^\d{4}-\d{2}-\d{2}$ Date (YYYY-MM-DD)
[\w\.-]+@[\w\.-]+ Email
https?://[^\s]+ URL
[A-Z][a-z]+ Capitalized word

6. Python re Functions

```
import re

# Search anywhere
```

Python Regex Cheat Sheet

```
re.search(r"cat", text)

# Match from start
re.match(r"cat", text)

# Find all matches
re.findall(r"\d+", text)

# Replace
re.sub(r"\s+", " ", text)

# Compile for reuse
pattern = re.compile(r"\d+")
pattern.findall(text)
```

7. Quick Visual Memory

Anchors: ^...$ = "from start to end"
Digits: \d = digit, \D = not digit
Words: \w = word char, \W = not word char
Spaces: \s = space, \S = not space
Repeat: * (0+), + (1+), ? (0/1), {n} (exact)
Choice: | = OR
Sets: [abc] = a or b or c, [^abc] = not a/b/c