

# Regex

David Arroyo Menéndez

November 11, 2019

# Summary

Some characters are special metacharacters, and don't match themselves. Instead, they signal that some out-of-the-ordinary thing should be matched. A full list of metacharacters:

`. ^ $ * + ? { } [ ] \ | ( )`

So, you can precede them with a backslash to remove their special meaning:

```
$ echo "^Lorem ipsum" > txt.txt
$ echo "Dolor" >> txt.txt
$ grep "\^" txt.txt
$ echo "\"" > txt.txt
$ echo "Dolor" >> txt.txt
$ grep "\^" txt.txt
$ grep "o|e" txt.txt
```

# Begin and End

```
$ grep "^Lorem" txt.txt
$ grep "^ipsum" txt.txt
$ grep "^Dolor" txt.txt
$ grep "Lorem$" txt.txt
$ grep "ipsum$" txt.txt
$ grep "Dolor$" txt.txt
```

# Repeating

```
$ echo "oooo" > txt.txt
$ egrep -i "o+" txt.txt
$ egrep -i "a+" txt.txt
$ egrep -i "a?" txt.txt
$ egrep -i "o{2,6}" txt.txt
```

## Metacharacters (IV). Grouping explicting

```
$ grep "[abc]" txt.txt  
$ grep "[rem]" txt.txt  
$ echo "niñas" >> txt.txt  
$ grep "niñ[oa]" txt.txt
```

# Metacharacters (V) Grouping with Ranges

```
$ grep "[a-c]" txt.txt
$ echo "13579" > numbers.txt
$ echo "2468" >> numbers.txt
$ echo "1456789" >> numbers.txt
$ grep "[0-1]" numbers.txt
```

## Metacharacters (VI). Grouping all

```
$ echo "davidam@gnu.org" | egrep -i ".*@.*"  
$ echo "davidam@gnu.org" | egrep -i "asdf@.*"
```



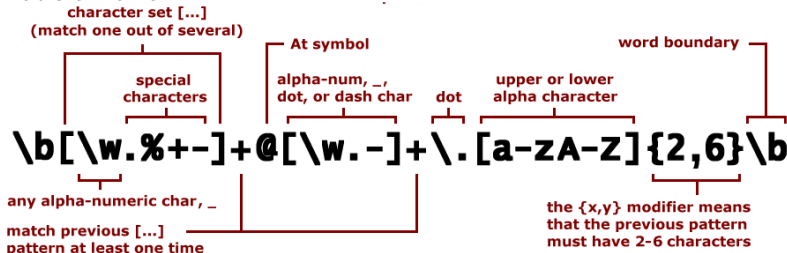
# Metacharacters (VI) Alternatives

```
$ uname -a | egrep -i "linux|unix"
```

# Metacharacters (III): Python metacharacters

# Metacharacters (IV): Explaining Python metacharacters

That's an email



**Parse: username@domain.TLD (top level domain)**

Si queremos encontrar un número, podemos usar una clase como <sup>1</sup>, o podemos utilizar un rango. Un rango es una clase de caracteres abreviada que se crea escribiendo el primer caracter del rango, un guión y el último caracter del rango. Múltiples rangos pueden definirse en la misma clase de caracteres.

---

<sup>1</sup>DEFINITION NOT FOUND.

```
$ python3 regex-basics.py  
$ python3 alfanum.py  
$ python3 ascii-encode.py  
$ python3 clean-string.py  
$ python3 date.py
```

# Capturing matches (I)

```
$ python3 group.py  
$ python3 sub.py  
$ python3 urls.py  
$ python3 unicode.py  
$ python3 hexadecimal.py  
$ python3 split.py  
$ python3 startend.py
```

## Capturing matches (II)

```
$ python3 findall.py  
$ python3 findall2.py  
$ python3 findall3.py  
$ python3 finditer.py  
$ python3 names.py  
$ python3 number.py  
$ python3 php.py
```

```
$ python3 infinitivos-participios.py  
$ python3 duplicate-lines.py  
$ python3 remove-blank-lines.py
```



Copyright (C) 2019 David Arroyo Menendez Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in GNU Free Documentation License.