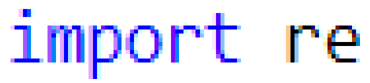
Modul 2

Regular Expression

Purwadhika
Startup and Coding School

Python Regular Expression (RegEx)

- Regular expression is a sequence of character with a purpose to matching with a given string



```
import re
```

- We implement the module by importing the re module

Set of function from Re module

- There are few functions that often used from Re module to search for a pattern in the string, they are:
  - search
  - findall
  - sub
  - split
  - compile

re.search

```
months = ['February', 'March' , 'April’ , 'May’ , 'June’ ]

for month in months:
    if re.search(r'a’ , month):
        print(month)
```

search function only be used to find if the regular expression match the string or not. After each string, it would stop after it find the word with the matched string

Metacharacter

Metacharacters are special character that did not match with themselves instead, they are used for signaling some patterns to be matched. The list are:

Metacharacter	Explanation	Example
.	Match any character (except newline)	"Ju.."
^	Match character that starts with these metacharacter	"^hello"
$	Match character that ends with these metacharacter	"world$"
*	Specifies that the previous character can be matched zero or more times, instead of exactly once	"ca*t"
+	Specifies that the previous character can be matched one or more times, instead of exactly once	"ca+t"
?	Specifies that the previous character can be matched zero or one times	"dash-?board"
{*m*,*n*}	Specifies that the previous character can be matched m times until n times	"ao{1,3}b"
|	Alternation, or the 'or' operator	'no|yes'
[ ]	A set of character to match with string	'[a-z]'
( )	Creating a group of character	'(ab)+c'

Sets

Set is a special character inside [ ]. It is used to match any string within the set of character written inside the [ ].

Set	Explanation
'[alp]'	Match character 'a' , 'l' , and 'p'
'[a-z]'	Match any lowercase word character between a to z
'[a-zA-Z]'	Match any word character between a to z
'[^a-z]'	Match any character except lowercase a to z
'[0123]'	Match character '0' , '1' , '2' , and '3'
'[0-9]'	Match any digit from 0 to 9
'[0-9][0-9]'	Match only specific number between 00 to 99
'[+$()]'	Special character lose their meaning if they located inside the set. It would match character '+' , '$', '(' , and ')'

Purwadhika
Startup and Coding School

re.findall

```
string = 'February March April May June'
match = re.findall(r'[a-zA-Z]+', string)
print(match)
```

findall is used to find every word that match the sequence that was given. In regex, word are defined as sequence of alphanumeric characters, so the end of a word is indicated by whitespace or a non-alphanumeric character.

Purwadhika
Startup and Coding School

Special Sequences

Special sequence are special character that have special meaning. The sequence always started by ‘\’ character and followed by one character. The character is listed below:

Character	Explanation
‘\w’	Match any alphanumeric character and underscore. It is almost equivalent to the set ‘[a-zA-Z0-9_]’
‘\W’	Match any non-alphanumeric character and underscore. It is almost equivalent to ‘[^a-zA-z0-9_]’
‘\s’	Match any whitespace character
‘\S’	Match any non-whitespace character
‘\d’	Match any decimal digit. It is equivalent to ‘[0-9]’
‘\D’	Match any non-decimal digit. It is equivalent to ‘[^0-9]’
‘\A’	Match if the character is at the start of the string. When not in the MULTILINE mode, ‘\A’ and ‘^’ are technically the same.
‘\Z’	Match if the character is at the end of the string. When not in the MULTILINE mode, ‘\Z’ and ‘$’ are technically the same.
‘\b’	Match if the character is at the start and/or at the end of the string
‘\B’	Match if the character is not at the start and/or at the end of the string

hika
ng School

re.sub

```
word = “Where 1 cat owner's is never been there!!"
print(re.sub( r‘[^a-zA-Z\']’ , ‘ ’ , word))
```

re.sub is used to replace character that match the pattern with the specific string to be replaced

re.split

```
word = "Only 2 time but 1 is enough. The ! is another sign"
print(re.split( r‘[0-9.!]’ , word))
```

re.split is used to split the character to become list by splitting it in the specified character sequence.

re.compile

In Python, creating a new regular expression pattern to match many strings can be slow, so it is recommended that you compile them if you need to be testing or extracting information from many input strings using the same expression. This method returns a re.RegexObject.

```
regex= re.compile(pattern)
```

The RegexObject could use previous function that have been covered before, such as search, findall, sub, and split

Purwadhika
Startup and Coding School

re.compile

```
regex = re.compile( ‘[a-zA-Z0-9]+’ )
word = "Do not feed the monkey's. It is really bad!!!"
print(regex.findall(word))
print(regex.sub('', word))
```

Purwadhika
Startup and Coding School