

Programming and Problem Solving through Python Language O Level / A Level

Chapter - 6 : Functions

Match Object

- A Match Object is an object containing information about the search.
- Match objects always have a boolean value of **True**.
- `match()` and `search()` function return **None** when there is no match, so we can use `if` statement, to check a match found.

Example

```
# Print the position (start- and end-position) of the first match occurrence
import re

txt = "398a811 356, 2102 1111"
x = re.search("11", txt)
print(x)

x = re.search("(\\d{3}) (\\d{3})", txt)
print(x) #this will print an Match Object

x = re.match("(\\d{3}) (\\d{3})", txt)
print(x) #this will print an Match Object

x = re.findall("(\\d{3}) (\\d{3})", txt)
print(x) #this will print an Match Object
```

Output

```
<re.Match object; span=(5, 7), match='11'>
<re.Match object; span=(4, 11), match='811 356'>
None
[('811', '356'), ('102', '111')]
```

Methods of Match Object

The Match object has properties and methods used to retrieve information about the search, and the result:

- `.span()` returns a tuple containing the start-, and end positions of the match.
- `.start()` returns the start position of match
- `.end()` returns the end position of the match
- `.re` returns the regular expression string
- `.string` returns the string passed into the function.
- `.group( num=0 )` returns the entire match (or specific subgroup num).
- `.groups()` returns all matching subgroups in a tuple.
- `finditer(pattern, string, flags=0)` Return an iterator with MatchObject instances over all non-overlapping matches for the RE pattern in string.

Example

#To search a group of 3 digits followed by group of 3 digits.

```
import re

txt = "398a811 356, 2102 1111"
x = re.search("(\\d{3}) (\\d{3})", txt)
print('Span -', x.span( ))
print('Span Start -',x.start( ))
print('Span End-',x.end( ))
print('String -', x.string)
print('Group -', x.group( ))
print('Groups -', x.groups( ))
print('Group(0) -', x.group(0))
print('Group Part(1) -', x.group(1))
print('Group Part(2) -', x.group(2))
print('Group Part(1,2) -', x.group(1,2))
print('Regular Exp -', x.re)
```

Output

```
Span - (4, 11)
Span Start - 4
Span End- 11
String - 398a811 356, 2102 1111
Group - 811 356
Groups - ('811', '356')
Group(0) - 811 356
Group Part(1) - 811
Group Part(2) - 356
Group Part(1,2) - ('811', '356')
Regular Exp - re.compile('(\\d{3}) (\\d{3}))')
```

Example

To search a group of 3 digits followed by group of 2 digits.

search() method able to match the pattern

match() method not able to match the pattern and return None.

```
import re
```

```
string = '398a01 356, 2102 1111'
```

Three digit number followed by space followed by two digit number

```
pattern = '(\\d{3}) (\\d{2})'
```

match variable contains a Match object.

```
match = re.search(pattern, string)
```

```
if match:
```

```
    print(match.group( ))
```

```

else:
    print("pattern not found")
match = re.match(pattern, string)

if match:
    print(match.group( ))
else:
    print("pattern not found")

```

Output

```

102 11
pattern not found

```

Example

```

import re

print('Output of Findall')
abc = 'ajay@google.com, vijay.kumar@hotmail.com, ajay_singh@yahoomail.com'
emails = re.findall(r'[\w\.-]+@[\w\.-]+', abc)
for email in emails:
    print(email)

print('Output of FindFilter')
abc = 'ajay@google.com, vijay.kumar@hotmail.com, ajay_singh@yahoomail.com'
emails = re.finditer(r'[\w\.-]+@[\w\.-]+', abc)
for email in emails:
    print(email.group( ))

```

Ouput

Output of Findall

```

ajay@google.com
vijay.kumar@hotmail.com
ajay_singh@yahoomail.com

```

Output of FindFilter

```

ajay@google.com
vijay.kumar@hotmail.com
ajay\_singh@yahoomail.com

```