Not much in these notes, please check the revision notes

1. Colab: https://colab.research.google.com/drive/1LAEJJ9-B0FyzCoy53balpGc7j2IhPBUl?usp=sharing
2. RegEx Practice: https://regexr.com/

```
!gdown 1sSDV5UspYZL3UUOGuiuxppSGcv1wS9ex
```

```
Downloading...
From: https://drive.google.com/uc?id=1sSDV5UspYZL3UUOGuiuxppSGcv1wS9ex
To: /content/data.txt
100% 9.33k/9.33k [00:00<00:00, 13.5MB/s]
```

```
data = open("data.txt", "r").read()
```

```
type(data)
```

```
str
```

```
print(data[:500])
```

```
Dave Martin
615-555-7164
173 Main St., Springfield RI 55924
davemartin@bogusemail.com

Charles Harris
800-555-5669
969 High St., Atlantis VA 34075
charlesharris@bogusemail.com

Eric Williams
                                47


Corey Jefferson
900-555-9340
826 Elm St., Epicburg NE 10671
coreyjefferson@bogusemail.com

Jennifer Martin-White
714-555-7405
212 Cedar St., Sunnydale CT 74983
jenniferwhite@bogusemail.com

Erick Davis
800-555-6771
519 Washington St.,
```

Saved successfully!

```
# masking email
```

```
def mask_email(s):
  if "@" in s:
    name, domain = s.split("@")
    return f"{name[0]}#######{name[-1]}@{domain}"
```

```
mask_email("abcd@efgh.com")
```

```
'a#######d@efgh.com'
```

```
mask_email("abcs.com") # invalid
```

```
mask_email("a@efgh.com") # invalid
```

```
'a#######a@efgh.com'
```

```
import re
```

```
def is_vemail(s):
  email_pattern = "^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$"
  res = re.search(email_pattern, s)
  if res:
    return True
  else:
    return False
```

```
.       - Any Character Except New Line
\d      - Digit (0-9)
\D      - Not a Digit (0-9)
\w      - Word Character (a-z, A-Z, 0-9, _)
\W      - Not a Word Character
                                        ewline)
                                        b, newline)

\b      - Word Boundary
\B      - Not a Word Boundary
^       - Beginning of a String
$       - End of a String

# Anchors
[]      - Matches Characters in brackets
[^ ]    - Matches Characters NOT in brackets
|       - Either Or
( )     - Group



# Quantifiers
```

Saved successfully!

```
 *       - 0 or More
 +       - 1 or More
 ?       - 0 or One
 {3}     - Exact Number
 {3,4}   - Range of Numbers (Minimum, Maximum)
```

```
def is_vemail(s):
  email_pattern = "^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$" # not readable
  res = re.search(email_pattern, s)
  if res:
    return True
  else:
    return False


regex_verbose = re.compile(r"""           # VERY readable and easy to understand. S
            ^\w+([\.-]?\w+)*              # start, \w+,
            @                             # single @ sign
            \w+([\.-]?\w+)*               # Domain name
            (\.\w{2,3})+$                 # .com, .ac.in,
             """,re.VERBOSE | re.IGNORECASE)   # no need to worry about these flags

res = regex_verbose.match("abcd@iisc.ac.in"); # no need to worry about the Python f
print(res.string)
print(res)
```

```
data
```

```
'Dave Martin\n615-555-7164\n173 Main St., Springfield RI 55924\ndavemartin@bog
rris@bogusemail.com\n\nEric Williams\n560-555-5153\n806 1st St., Faketown AK 8
urg NE 10671\ncoreyjefferson@bogusemail.com\n\nJennifer Martin-White\n714-555-
00-555-6771\n519 Washington St., Olympus TN 32425\ntomdavis@bogusemail.com\n\n
com\n\nLaura Jefferson\n516-555-4615\n890 Main St., Pythonville LA 29947\nlaur
                     mail.com\n\nMichael Arnold\n608-555-4938\n249 Elm
```

Saved successfully! ×

1. match : Checks for a match only at the beginning of the string
2. search : Locates the pattern in the string
3. findall : Find all occurence of the string
4. finditer: Return an iterator yielding match objects over all non-overlapping matches

```
# extrat phone numbers
pattern ="\d{3}-\d{3}-\d{4}"
print(re.match(pattern, data))
```

```
None
```

```
# extrat phone numbers
pattern ="\d{3}-\d{3}-\d{4}"
print(re.search(pattern, data))
```

```
    <re.Match object; span=(12, 24), match='615-555-7164'>
```

```
# extrat phone numbers
pattern = "\d{3}-\d{3}-\d{4}"
print(re.findall(pattern, data))
```

```
    ['615-555-7164', '800-555-5669', '560-555-5153', '900-555-9340', '714-555-7405
```

```
# extract phone numbers
pattern ="\d{3}-\d{3}-\d{4}"
nums = re.finditer(pattern, data)
for i, num in enumerate(nums):
  print(num)
  if i == 5:
    break
```

```
    <re.Match object; span=(12, 24), match='615-555-7164'>
    <re.Match object; span=(102, 114), match='800-555-5669'>
    <re.Match object; span=(191, 203), match='560-555-5153'>
    <re.Match object; span=(281, 293), match='900-555-9340'>
    <re.Match object; span=(378, 390), match='714-555-7405'>
    <re.Match object; span=(467, 479), match='800-555-6771'>
```

```
# extract phone numbers
pattern ="\d{3}-\d{3}-\d{4}"
nums = re.finditer(pattern, data)
for i, num in enumerate(nums):
  print(num.start(), num.end(), num.group())
  if i == 5:
    break
```

Saved successfully!

```
    191 203 560-555-5153
    281 293 900-555-9340
    378 390 714-555-7405
    467 479 800-555-6771
```

```
# extract emails
pattern = "\w+@\w+.\w{2,3}"
emails = re.finditer(pattern, data)
for i, email in enumerate(emails):
  print(email.start(), email.end(), email.group())
  if i == 5:
    break
```

```
    60 85 davemartin@bogusemail.com
    147 175 charlesharris@bogusemail.com
    235 263 laurawilliams@bogusemail.com
    325 354 coreyjefferson@bogusemail.com
    425 453 jenniferwhite@bogusemail.com
    517 540 tomdavis@bogusemail.com
```

```
pattern = '\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+'
emails = re.finditer(pattern, data)
for i, email in enumerate(emails):
  print(email.start(), email.end(), email.group())
  if i == 5:
    break
```

```
    60 85 davemartin@bogusemail.com
    147 175 charlesharris@bogusemail.com
    235 263 laurawilliams@bogusemail.com
    325 354 coreyjefferson@bogusemail.com
    425 453 jenniferwhite@bogusemail.com
    517 540 tomdavis@bogusemail.com
```

```
# Extract Names
```

```
pattern = "[A-Z][a-z]*\s[A-Z][a-z]{2,}"
names = re.finditer(pattern,data)
for i, name in enumerate(names):
  print(name)
  if i == 5:
    break
```

```
    <re.Match object; span=(0, 11), match='Dave Martin'>
    <re.Match object; span=(87, 101), match='Charles Harris'>
    <re.Match object; span=(177, 190), match='Eric Williams'>
    <re.Match object; span=(265, 280), match='Corey Jefferson'>
    <re.Match object; span=(356, 371), match='Jennifer Martin'>
    <re.Match object; span=(455, 466), match='Erick Davis'>
```

```
regex_verbose = re.compile(r"""           # VERY readable and easy to understand. S
                                          # start, \w+,
                                          # single @ sign
            \w+([\.-]?\w+)*               # Domain name
            (\.\w{2,3})+$                 # .com, .ac.in,
             """,re.VERBOSE | re.IGNORECASE)    # no need to worry about these flags

res = regex_verbose.match("abcd@iisc.ac.in"); # no need to worry about the Python f
print(res.string)
print(res)
```

Saved successfully!

```
    abcd@iisc.ac.in
    <re.Match object; span=(0, 15), match='abcd@iisc.ac.in'>
```

```
re.search("a+", "aaaAAA")
```

```
    <re.Match object; span=(0, 3), match='aaa'>
```

```
re.search("A+", "aaaAAA")
```

```
    <re.Match object; span=(3, 6), match='AAA'>
```

```
re.search("[aA]+", "aaaAAA")
```

```
    <re.Match object; span=(0, 6), match='aaaAAA'>
```

```
re.search("a+", "aaaAAA", re.IGNORECASE)
```

```
    <re.Match object; span=(0, 6), match='aaaAAA'>
```

```
re.search("a+", "aaaAAA", re.I)
```

```
    <re.Match object; span=(0, 6), match='aaaAAA'>
```

```
# re.VERBOSE = re.X
# re.ASCII
```

```
target_str = "Priya is an Instructor at Scaler and her salary is 100000"
```

```
pattern = "^([A-Z]\w{2,}).+(\d{6,})$"
result = re.match(pattern, target_str)
result
```

```
    <re.Match object; span=(0, 57), match='Priya is an Instructor at Scaler and he
```

```
result.start(), result.end(), result.group()
```

```
    (0, 57, 'Priya is an Instructor at Scaler and her salary is 100000')
```

```
result.group(1)
```

Saved successfully! ×

```
result.group(2)
```

```
    '100000'
```

```
result.group(0)
```

```
    'Priya is an Instructor at Scaler and her salary is 100000'
```

```
pattern = r'(\w+)@(\w+)\.(\w{2,3})'
emails = re.finditer(pattern, data)
for i, email in enumerate(emails):
  print(email.group(), email.group(1), email.group(2), email.group(3))
  if i == 5: # printing first five
    break
    davemartin@bogusemail.com davemartin bogusemail com
    charlesharris@bogusemail.com charlesharris bogusemail com
    laurawilliams@bogusemail.com laurawilliams bogusemail com
    coreyjefferson@bogusemail.com coreyjefferson bogusemail com
    jenniferwhite@bogusemail.com jenniferwhite bogusemail com
    tomdavis@bogusemail.com tomdavis bogusemail com
```

```
def mask_email(s):
  if "@" in s:
    name, domain = s.split("@")
    return f"{name[0]}#######{name[-1]}@{domain}"
```

```
pattern = '\w+@\w+.[a-z]{3}'
emails = re.findall(pattern,data)
print(emails)
```

```
    ['davemartin@bogusemail.com', 'charlesharris@bogusemail.com', 'laurawilliams@b
```

```
for email in emails:
  print(mask_email(email))
```

```
    e#######s@bogusemail.com
    m#######s@bogusemail.com
    l#######s@bogusemail.com
    d#######e@bogusemail.com
    l#######s@bogusemail.com
    s#######e@bogusemail.com
    l#######n@bogusemail.com
    c#######n@bogusemail.com
    j#######n@bogusemail.com
    m#######n@bogusemail.com
    c#######r@bogusemail.com
    j#######e@bogusemail.com
    j#######t@bogusemail.com
    c#######n@bogusemail.com
    j#######s@bogusemail.com
```

Saved successfully!

```
    c#######n@bogusemail.com
    s#######s@bogusemail.com
    p#######s@bogusemail.com
    j#######s@bogusemail.com
    p#######n@bogusemail.com
    b#######s@bogusemail.com
    j#######r@bogusemail.com
    b#######s@bogusemail.com
    t#######n@bogusemail.com
    s#######n@bogusemail.com
    s#######n@bogusemail.com
    m#######n@bogusemail.com
    s#######n@bogusemail.com
    c#######s@bogusemail.com
    l#######n@bogusemail.com
    t#######s@bogusemail.com
    p#######r@bogusemail.com
    b#######s@bogusemail.com
    m#######n@bogusemail.com
```

```
m#######n@bogusemail.com
k#######r@bogusemail.com
n#######t@bogusemail.com
l#######n@bogusemail.com
c#######s@bogusemail.com
j#######n@bogusemail.com
c#######r@bogusemail.com
r#######s@bogusemail.com
t#######n@bogusemail.com
t#######n@bogusemail.com
l#######s@bogusemail.com
n#######d@bogusemail.com
l#######n@bogusemail.com
j#######n@bogusemail.com
n#######e@bogusemail.com
m#######d@bogusemail.com
j#######s@bogusemail.com
m#######n@bogusemail.com
j#######t@bogusemail.com
r#######s@bogusemail.com
j#######r@bogusemail.com
j#######t@bogusemail.com
c#######r@bogusemail.com
```

```
pattern = "\d{3}-\d{3}-\d{4}"
nums = re.findall(pattern,data)
print(nums)
```

```
['615-555-7164', '800-555-5669', '560-555-5153', '900-555-9340', '714-555-7405
```

```
def mask_phone(p):
  if len(p) == 12:
    return f"###-###-{p[-3:]}"


print([mask_phone(num) for num in nums])
```

```
', '###-###-153', '###-###-340', '###-###-405', '#
```

Saved successfully! ×

✓ 0s completed at 23:37 ● ×

Saved successfully! ×