17/06/2022

WAP to get the below output -

Sentence = "hello world welcome to python programming hi there"

o/p → d = { 'h' : ['hello', 'hi'], 'w' : ['world', 'welcome'] }
....

by using normal method →

```
d={}
for word in Sentence.split():
    if word[0] not in d:
        d[word[0]] = [word]
    else:
        d[word[0]].append(word)
```

by using default dict →

```
dd = defaultdict(list)
for word in Sentence.split():
    dd[word[0]].append(word)
```

(or)

```
dd[word[0]] += [word]
```

Reverse the values in the dictionary if the value is of type string.

d = { 'a' : 'hello', 'b' : 100, 'c' : 10.1, 'd' : 'world' }

by using normal →

```
for key, value in d.items():
    if isinstance(value, str):
        d[key] = value[::-1]
    else:
        d[key] = value.
```

Comprehension →

d = { key : value[::-1] if instance(value, dict) else value
for (key, value in d.items()) }

Write a program to replace value present in nested dictionary.

replace 'note' with 'net'.

d = { 'a': 100, 'b': { 'm': 'man', 'n': 'nose', 'o': 'ox',
'c': 'cat' } }

```
def replace_dict(old_, new_, dict_):
    for key, value in dict_.items():
        if isinstance(value, dict):
            for k, v in value.items():
                if v == old_:
                    value[k] = new_
    return dict_
```

Group flowers and animals in the below list.

items = ['lotus-flower', 'lilly-flower', 'cat-animal',
'sunflower-flower', 'dog-animal']

String-argument
{ Substring, Start index, End index }

```
d = {}
for item in items:
    name, grp = item.split("-")
    if grp not in d:
        d[grp] = [name]
    else:
        d[grp].append(name)
```

① Grouping files with same extensions -

files = ['apple.txt', 'yahoo.pdf', 'gmail.pdf', 'google.txt', 'amazon.pdf', 'facebook.txt', 'flipkart.pdf']

② Map to get the indices of each item in the below list

names = ["apple", 'google', apple, yahoo, google, gmail, gmail, gmail]

output should →

PATTERN PROGRAMS

① for i in range(5):
    for j in range (i+1):
        Print ("*", end="")
    Print ()

(left justified triangle)

II nd Method → for row in range (1,6)
    Print ("*" * row)

② for row in range (6,0,-1):
    Print ("*" * row)

(5)

③ for row in range (1,6):
    Print (f" ("*" * row :>10)")

(5)

④ for row in range (6,0,-1):
    Print (f" ("*" * row :>12)")

(5)

⑤
```
for row in range (1,6)
    print (f"{'* ' * row:^10}")
```

(5)

⑥
```
for i in range (1,6):
    for j in range (1, i+1):
        print (j, end=" ")
    print ()
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

2nd Method →
```
pat = ""
for row in range (1,6):
    pat = pat + str(row) + " "
print (pat)
```

Assignment

```
1 2 3 *
1 2 * 4
1 * 3 4
* 2 3 4
```