This notebook will show all about Dictonaries. Dictonaries are like Tuple. But Tuple are disorder, Dictonaries are in order.

```
eng2sp = {'three':'tres', 'one':'uno', 'two':'dos'} #This line show English to spanis lang
value = eng2sp['two'] #value is the spanish for english Two
print(value)
print(eng2sp['one'])
```

```
dos
uno
```

Basic operations with Dictonaries

```
inventory = {'apples':430, 'bananas':312, 'oranges':525, 'pears':217} #Delclearing a Dicto
print(inventory)
del inventory['pears'] #delete a value
print(inventory)
inventory ['apples'] = 500 #update the value of apples
print(inventory)
inventory['bananas'] = inventory['bananas'] + 200 #update the value
print(inventory)
print(len(inventory))
```

```
{'apples': 430, 'bananas': 312, 'oranges': 525, 'pears': 217}
{'apples': 430, 'bananas': 312, 'oranges': 525}
{'apples': 500, 'bananas': 312, 'oranges': 525}
{'apples': 500, 'bananas': 512, 'oranges': 525}
3
```

Dictonaries Methods

```
for key in inventory:
  print(key, "has the value", inventory[key])
```

```
apples has the value 500
bananas has the value 512
oranges has the value 525
```

```
print(list(inventory.values()))
```

```
[500, 512, 525]
```

Aliasing and coping with dictonaries

```
opposites = {'up':'down', 'right':'wrong', 'true':'false'}
alias = opposites
print(alias == opposites)
alias['right'] = 'left'
print(opposites['right'])
print(opposites['right'])
```

```
True
left
```

Now suppose we want to keep track of how many 't' and 's' are in a file

```
f = open ('sample_data/README.md','r')
txt = f.read()
t_count = 0
s_count = 0
for c in txt:
  if c == 't':
    t_count = t_count + 1
  elif c == 's':
    s_count = s_count + 1
print('t:' + str(t_count)+' occurence')
print('s:' + str(s_count)+' occurence')
```

```
t:52 occurence
s:56 occurence
```

Create a dictonary called 'd' that keeps track of all the characters in the string 'placement' and notes how many times each character was seen. Then, find the key with the lowest value in this dictonary and assign that key to 'min_value'

```
placement = 'Because are cool places to visit in spring however the mackinaw Bridge is nea
d = {}
for c in placement:
  if c not in d:
    d[c] = 0
    d[c] = d[c]+1
keys = list(d.keys())
min_value = key[0]

for key in keys:
  if d[key] < d[min_value]:
    min_value = key
print(min_value)
```

```
o
```