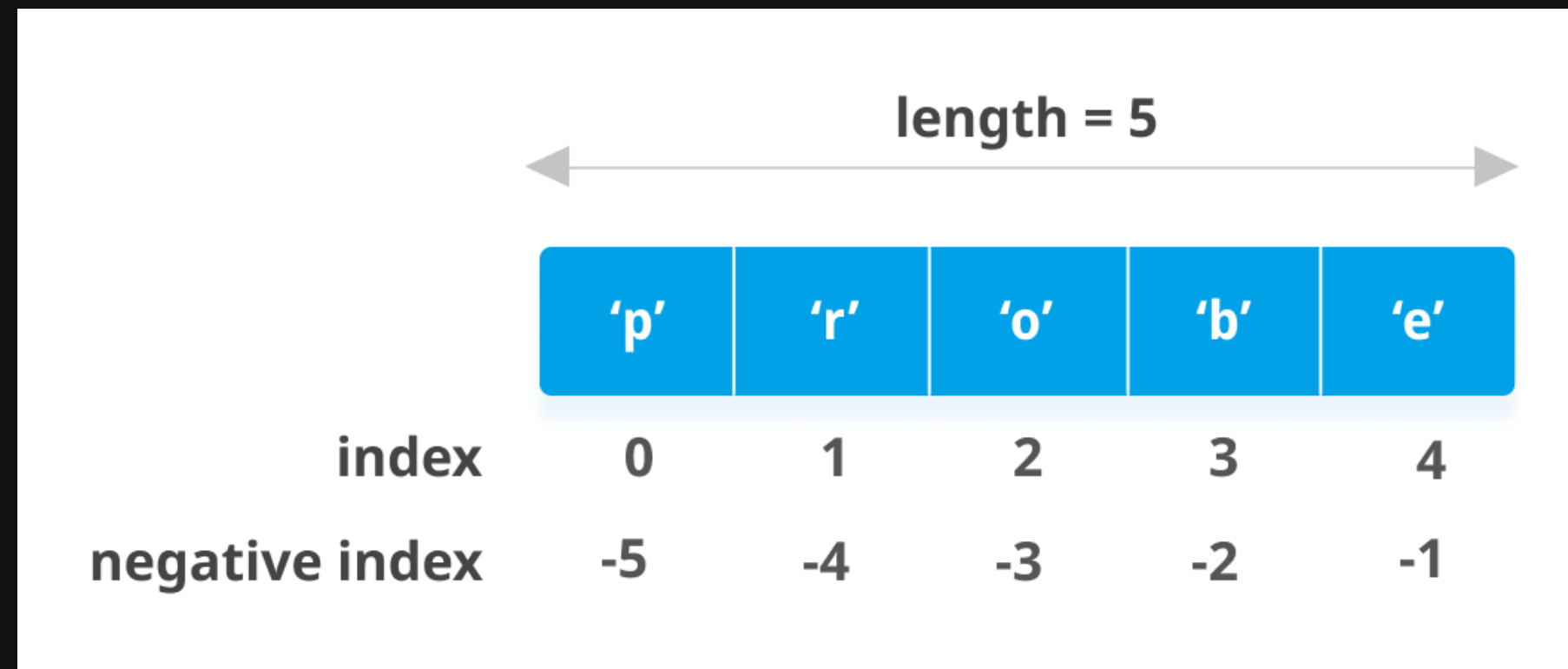


COMP1531

2.2 - Python - Dictionaries

Python - Dictionaries

Lists are **sequential containers** of memory. Values are referenced by their **integer index** (key) that represents their location in an **order**



Python - Dictionaries

Dictionaries are **associative containers** of memory. Values are referenced by their **string key** that *maps* to a value

name —————→ "sally"

age —————→ 18

height —————→ "187cm"

Python - Dictionaries

Dictionaries are **associative containers** of memory. Values are referenced by their **string key** that *maps* to a value

dict_basic_1.py

```
1 userData = {}  
2 userData["name"] = "Sally"  
3 userData["age"] = 18  
4 userData["height"] = "187cm"  
5 print(userData)
```

```
1 {'name': 'Sally', 'age': 18, 'height': '187cm'}
```

Python - Dictionaries

There are a number of different ways
we can construct and interact with
dictionaries

dict_basic_2.py

```
1 userData = {  
2     'name' : 'Sally',  
3     'age' : 18,  
4     'height' : '186cm', # Why a comma?  
5 }  
6 userData['height'] = '187cm'  
7 print(userData)
```

```
1 {'name': 'Sally', 'age': 18, 'height': '187cm'}
```

Python - Dictionaries

dict_loop.py

Basic loops are over **keys**
not **values**:

How would we modify this
to print out the values
instead?

```
1  userData = [  
2      {  
3          'name' : 'Sally',  
4          'age' : 18,  
5          'height' : '186cm',  
6      }, {  
7          'name' : 'Bob',  
8          'age' : 17,  
9          'height' : '188cm',  
10     },  
11 ]  
12 for user in userData:  
13     print("Whole user: ", user)  
14     for part in user:  
15         print(f" {part}")
```

```
1 Whole user: {'name': 'Sally', 'age': 18, 'height': '186cm'}  
2   name  
3   age  
4   height  
5 Whole user: {'name': 'Bob', 'age': 17, 'height': '188cm'}  
6   name  
7   age  
8   height
```

Python - Dictionaries

dict_loop_2.py

```
1 userData = {'name' : 'Sally', 'age' : 18, \  
2             'height' : '186cm'}  
3  
4 for user in userData.items():  
5     print(user)  
6 print("=====  
7  
8 for user in userData.keys():  
9     print(user)  
10  
11 print("=====  
12 for user in userData.values():  
13     print(user)
```

```
1 ('name', 'Sally')  
2 ('age', 18)  
3 ('height', '186cm')  
4 =====  
5 name  
6 age  
7 height  
8 =====  
9 Sally  
10 18  
11 186cm
```

Python - Dictionaries

Optional Activity

Q. Write a python program that takes in a series of words from STDIN and outputs the frequency of how often each vowel appears