# String operations

There are different ways you can modify a string

For instance, you can make it all uppercase

name= “john smith”

print(name.upper())

message = “ThIs SeNtEnCe ShOuLd Be UpPerCaSe”

print(message.upper())

- We can also make it all lowercase
- name= “JOHN SMITH”
- print(name.lower())
- message = “ThIs SeNtEnCe ShOuLd Be LoWeRcAsE”
- print(message.lower())
- greeting= “HELLO”
- lowercase_greeting= greeting.lower()
- print(lowercase_greeting)

# Print part of string – indexing and slicing

- Can also print a part of the string
- To do this you have to count the index of the characters where you want to start from and where you want to end
- The first character is always index 0 so start counting from 0, not 1

| H | E | L | L | O |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

- **greeting="HELLO"**
- **print(greeting[0])**
- This will print H
- **print(greeting[-1])**
- This will print O (last letter)
- **print(greeting[0:2]**
- This will print HE
- **print (greeting[1:4])**
- Will print ELL

| I | | L | i | k | e | | C | o | d | i | n | g |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

- **message=“I Like Coding”**
- What would **print(message[0:5])** print out?
- What would **print(message[3:9])** print out?
- How would you print out “Like” using message?
- How would you print out “Codi”?

- Can also skip characters
- **greeting=“HelloHello”**
- **print(greeting[0:5:2])** will print **Hlo**. So it prints every 2 characters each time from character 0 up to but not including 5
- [start:stop:step]
- First number is the character number you start at (count from 0)
- Second number is character after the one you want to stop at
- Will print up to but not including that character
- Third number is how you want it to skip.
- **print(greeting[::2])** will print **Hloel**. Leaving the first 2 blank means you want it to apply to the entire string.

| H | e | l | l | o | H | e | l | l | o |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

- Can also reverse string.
- **message = “Goodbye”**
- **print(message[::-1])** will print **eybdooG**
- First 2 are blank which means it applies to the whole string
- -1 as the step means start from last character and print in reverse

- Can also split a string on space
- weekdays="Monday Tuesday Wednesday Thursday Friday"
- daylist=weekdays.split()
- print(daylist)
- This splits weekdays on space and creates a list of weekdays
- Prints ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']
- So each day is now separated
- We can then print each element in the list separately also by using their index

| Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

- print(daylist[1]) will print Tuesday

# Combine 2 strings

- You can take 2 strings and combine them into 1
- Use + to combine
- **word1=“Hello”**
- **word2=“World”**
- **greeting = word1+word2**
- **print(greeting)**
- Can’t combine a string and a number
- **name=“John”**
- **num=5**
- **print(name+num)**
- This will give an error
- But if you change it to **num=“5”** it will work

# Get length of string

string1 = “Apples”

print(len(string1)) #will print 6

len() returns the length of the string.

# Replace part of a string

```
string1 = “I like cats”
string2 = string1.replace(“cats”, “dogs”)
print(string2) #will print I like dogs
```

# Count how many times a character appears

```
string1=“banana”
print(string1.count(“a”)) #will print 3
```

# Review

- a = “My name is John”
- a.upper() #changes to all uppercase
- a.lower() #changes to all lowercase
- len(a) #returns number of characters in string
- a[0] #returns 1st character (index 0)
- a[-1] #returns last character
- a[2:5] #returns characters from 2nd up to but not including 5th index
- a[2:10:2] #returns every second character within that range (start:stop:step)
- a.split() #returns a list of each word in string (splits it on each space)
- a.replace(“John”, “Steve”) #replaces John with Steve
- a.count(“a) #returns number of a’s in the string

# Program 1: Change message

- Ask the user to write in a message
- Print the message in all uppercase
- Print the message in all lowercase
- Print the first 5 characters of the message
- Print every 3 characters in the message
- Print the message backwards

# Program 2: First and last name

- Ask the user to enter their first and last name separated by a space like this: “John Smith”
- Split it into first and last name
- Print out the first and last name separately
- Print out like this First name: What the user put in
- Last name: What the user put in

# Program 3: Initials

- Now prints the user’s initials
- Ex. John Smith should print J.S.

# Program 4: Create a palindrome

- A palindrome is a word or phrase that’s the same forward and backward like racecar
- Ask the user to enter a word or phrase
- Reverse the phrase the user entered
- Combine the user’s phrase with the reversed phrase
- Print out the result
- This should create a palindrome

# Program 5: Count letters

- Ask the user to enter a word or phrase
- Ask the user to enter a letter
- Print how many times that letter appears in the user’s word or phrase

# Censor word

- Ask the use to enter a phrase.
- Ask them to enter a word in the phrase they want to censor
- Replace every instance of that word with **** or [censored]