Define strings?

- IN programming strings are sequences of characters used to represent text. A string is a sequence of characters enclosed in either single quotes ('') or double quotes (“”) depending on the programming language.strings can include letters, number, symbols and whitespace.

Representation of strings?

- Basic string Representation using single or double quotes to create a string.

```
string1 = "Hello, World!"
string2 = 'Python is fun!'
print(string1)  #Output: Hello, world!
print(string2)  #Output: Python, is fun!
```

```
Hello, World!
Python is fun!
```

*2. Multiline string using triple quotes for a multiline string.

```
multiline_string = """my name is kolluri gopi chowdary."""
print(multiline_string)
# Output: kolluri gopi chowdary.
#.
```

```
my name is kolluri gopi chowdary.
```

1. String concatenation concat
1. Escape characters in string using escape sequences like\n for newlines or\t for tabs.

```
string_with_escape = "Line 1\nLine 2\n\tIndented Line"
print(string_with_escape)
# Output:
# Line 1
# Line 2
#     Indented Line
```

```
Line 1
Line 2
     Indented Line
```

Apply string operator examples for each 5 times to repetition

```
1.Concatenation (+ Operator)
Concatenating a string five times in a loop.

base_string = "mumbai indians"
result = ""
for i in range(5):
    result += base_string + " "
print(result)  # Output:  mumbai indians mumbai indians mumbai indians mumbai indians mumbai indians

mumbai indians mumbai indians mumbai indians mumbai indians mumbai indians

for i in range(5):
    result = "Count: " + str(i)
    print(result)
# Output:
# Count: 0
# Count: 1
# Count: 2
# Count: 3
# Count: 4

Count: 0
Count: 1
Count: 2
Count: 3
Count: 4

name = "kolluri karthik"
for i in range(5):
    greeting = "Hi, " + name + "!"
    print(greeting)

Hi, kolluri karthik!
Hi, kolluri karthik!
Hi, kolluri karthik!
Hi, kolluri karthik!
Hi, kolluri karthik!

word1 = "Good"
word2 = "evening"
for i in range(5):
    message = word1 + " " + word2
    print(message)
```

```
Good evening
Good evening
Good evening
Good evening
Good evening
```

```
for i in range(5):
    combined_string = "Number: " + str(i)
    print(combined_string)
```

```
Number: 0
Number: 1
Number: 2
Number: 3
Number: 4
```

```
   Repetition (* Operator)
Using the multiplication operator to repeat a string five times.
```

```
  File "<ipython-input-17-0194eb68b8f5>", line 2
    Using the multiplication operator to repeat a string five times.
          ^
SyntaxError: invalid syntax
```

```
base_string = "devara "
result = base_string * 5
print(result)  # Output: devara devara devara devara devara
```

```
devara devara devara devara devara
```

```
phrase = " I told you Go! "
for i in range(5):
    print(phrase * (i+1))
```

```
 I told you Go!
 I told you Go!  I told you Go!
 I told you Go!  I told you Go!  I told you Go!
 I told you Go!  I told you Go!  I told you Go!  I told you Go!
 I told you Go!  I told you Go!  I told you Go!  I told you Go!  I
told you Go!
```

```
char = "*"
for i in range(5):
    print(char * (i+1))
```

```
*
**
***
****
*****
```

```python
number_str = "420"
for i in range(5):
    print(number_str * (i+1))
```

```
420
420420
420420420
420420420420
420420420420420
```

```python
string = "Repeat "
for i in range(5):
    print(f"{i+1}: {string * (i+1)}")
# Output:
# 1: Repeat
# 2: Repeat Repeat
# 3: Repeat Repeat Repeat
# 4: Repeat Repeat Repeat Repeat
# 5: Repeat Repeat Repeat Repeat Repeat
```

```
1: Repeat
2: Repeat Repeat
3: Repeat Repeat Repeat
4: Repeat Repeat Repeat Repeat
5: Repeat Repeat Repeat Repeat Repeat
```

Create a string, take a paragraph access index 21,47,105,129,201,241,311,395,409,418?

```python
# Step 1: Store the paragraph as a string
paragraph = """
The world of technology has rapidly evolved over the past few decades, transforming how we live, work, and interact. From the introduction of the personal computer to the explosion of the internet, we have witnessed a digital revolution that has changed the face of nearly every industry. One of the most significant advancements has been the rise of artificial intelligence (AI) and machine learning. These technologies have given machines the ability to learn from data, make decisions, and perform tasks that previously required human intervention. Whether it's recommending products online, predicting diseases in healthcare, or even driving cars autonomously, AI is becoming an integral part of everyday life.

Robotics is another fascinating area that has made tremendous strides in recent years. What was once considered the realm of science fiction is now a reality, with robots being used in manufacturing, healthcare, and even households. Robots equipped with advanced sensors and AI are
```

```
able to perform delicate tasks like surgery, while industrial robots can assemble cars with unmatched precision and speed. The combination of robotics and AI, often referred to as "cognitive robotics," is pushing the boundaries of what machines can achieve, leading to a future where robots will not only assist but also collaborate with humans in complex tasks.

The internet of things (IoT) has also played a pivotal role in modern technological advancements. By connecting everyday objects to the internet, IoT allows for a more integrated and efficient world. Smart homes, for instance, allow people to control lighting, temperature, and security systems remotely using their smartphones. In the industrial sector, IoT devices monitor machinery, predict failures, and improve efficiency. This connectivity extends to cities as well, where smart infrastructure improves transportation, energy usage, and waste management. The IoT is revolutionizing the way we interact with the physical world, making it smarter, more efficient, and more responsive to our needs.

Another transformative technology is blockchain, initially made famous by cryptocurrencies like Bitcoin. Blockchain is essentially a decentralized, distributed ledger that ensures transparency and security in transactions. It has applications far beyond digital currency, from securing voting systems to tracking supply chains and intellectual property rights. The use of blockchain technology in finance, healthcare, and even government has the potential to enhance security, reduce fraud, and build trust in digital systems. As businesses and governments recognize its benefits, blockchain is likely to become a staple in the infrastructure of tomorrow's digital economy.

While these technologies hold great promise, they also present challenges. Ethical concerns regarding privacy, security, and job displacement need to be addressed as AI and automation take over more tasks traditionally performed by humans. As machines become more intelligent, we must carefully consider how to balance innovation with ethical responsibility. Governments, businesses, and academia need to collaborate to create guidelines that ensure technological advancements benefit society as a whole, rather than exacerbating inequality.
"""

# Step 2: Access specific characters using indexing

print(paragraph[21])
print(paragraph[47])
print(paragraph[105])
print(paragraph[129])
print(paragraph[201])
print(paragraph[241])
```

```
print(paragraph[311])
print(paragraph[395])
print(paragraph[409])
print(paragraph[418])
```

```
o
e
n
t

t
f
a
t
i
```

slicing operators 5 examples each?

```
# Basic Slicing Examples
text = "Hello, World!"
print(text[2:6])  # Output: llo,

text = "Python Programming"
print(text[:5])  # Output: Python

text = "Data Science"
print(text[7:])  # Output: Science

text = "Artificial Intelligence"
print(text[3:9])  # Output: ifici

text = "Machine Learning"
print(text[5:])  # Output: ine Learning
```

```
llo,
Pytho
ience
ificia
ne Learning
```

```
# Slicing with Step Examples
text = "Programming"
print(text[0:10:2])  # Output: Prgaig

text = "Slicing Examples"
print(text[::3])  # Output: Sxme

text = "Python Slicing"
print(text[4:12:2])  # Output: o  i
```

```python
text = "Reverse"
print(text[::-1])  # Output: esreveR

text = "Step Slicing"
print(text[::-2])  # Output: gniclS tep
```

```
Pormi
Scgxps
o lc
esreveR
giiSpt
```

```python
# Slicing with Negative Indices Examples
text = "Python Coding"
print(text[-4:])  # Output: Coding

text = "Machine Learning"
print(text[-10:-5])  # Output: Lear

text = "Artificial Intelligence"
print(text[:-7])  # Output: Artificial Intellig

text = "Data Analysis"
print(text[-9:])  # Output: Analysis

text = "Advanced Slicing"
print(text[-1:-15:-2])  # Output: gnScv
```

```
ding
e Lea
Artificial Intel
 Analysis
giiSdca
```

```python
# Combining Slicing with Indexes Examples
text = "Exploration"
print(text[3:8][::-1])  # Output: ralo

text = "Artificial Intelligence"
middle = text[6:16]
print("Middle slice: " + middle)  # Output: Middle slice: icial Int

text = "String Manipulation"
print(text[7:14][1])  # Output: n

text = "Data Science"
print(text[2:11:3])  # Output: a c

text = "Python Programming"
print(text[7:16][::-1])  # Output: gnir
```

```
tarol
Middle slice: cial Intel
a
tSe
immargorP
```

explain string format and give 3 examples each?

```
Old-Style String Formatting (Using %)
This method uses % to format strings. It is an older style but still
widely used.

#Example 1: Displaying project details
project_name = "Robot Arm"
completion_year = 2024
formatted_string = "Project: %s, Completion Year: %d" % (project_name,
completion_year)
print(formatted_string)

# Example 2: Showing progress in machine learning
model_name = "DeepNet"
accuracy = 92.5
formatted_string = "Model: %s, Accuracy: %.2f%%" % (model_name,
accuracy)
print(formatted_string)

# Example 3: Reporting on robotics project stages
stage = "Prototype"
progress = 75
formatted_string = "Current Stage: %s, Progress: %d%%" % (stage,
progress)
print(formatted_string)

Project: Robot Arm, Completion Year: 2024
Model: DeepNet, Accuracy: 92.50%
Current Stage: Prototype, Progress: 75%
```

*str.format() Method Introduced in Python 2.6/3.0, this method allows more flexibility with positional and keyword arguments. "string {}".format(values) represented.

```
# Example 1: Detailed robotics project report
project_name = "Reddy"
duration_months = 18
formatted_string = "Project Name: {}, Duration: {}
months".format(project_name, duration_months)
print(formatted_string)

# Example 2: Machine learning model evaluation
```

```
model_name = "Neural Network"
precision = 89.7
formatted_string = "Model: {0}, Precision: {1:.1f}
%".format(model_name, precision)
print(formatted_string)

# Example 3: Robotics system integration status
system_name = "Vision System"
status = "Operational"
errors = 3
formatted_string = "System: {system_name}, Status: {status}, Errors
Found: {errors}".format(system_name=system_name, status=status,
errors=errors)
print(formatted_string)
```

```
Project Name: Reddy, Duration: 18 months
Model: Neural Network, Precision: 89.7%
System: Vision System, Status: Operational, Errors Found: 3
```

```
3) f-Strings (Formatted String Literals)
Introduced in Python 3.6, f-strings provide a way to embed expressions
inside string literals, using {} braces.
f"string {expression}"

# Example 1: Reporting on AI research progress
research_topic = "AI in Robotics"
papers_published = 5
formatted_string = f"Research Topic: {research_topic}, Papers
Published: {papers_published}"
print(formatted_string)

# Example 2: Displaying robot specifications
robot_model = "X-Robot 3000"
battery_life = 24  # in hours
formatted_string = f"Model: {robot_model}, Battery Life:
{battery_life} hours"
print(formatted_string)

# Example 3: Showing performance metrics of an AI system
accuracy = 95.3
training_time = 12  # in hours
formatted_string = f"System Accuracy: {accuracy:.1f}%, Training Time:
{training_time} hours"
print(formatted_string)
```

```
Research Topic: AI in Robotics, Papers Published: 5
Model: X-Robot 3000, Battery Life: 24 hours
System Accuracy: 95.3%, Training Time: 12 hours
```

String methods with each 1 example.

```
# 1. str.upper()
text = "robotics"
uppercase_text = text.upper()
print("Uppercase:", uppercase_text)

# 2. str.lower()
text = "ARTIFICIAL INTELLIGENCE"
lowercase_text = text.lower()
print("Lowercase:", lowercase_text)

# 3. str.capitalize()
text = "deep learning"
capitalized_text = text.capitalize()
print("Capitalized:", capitalized_text)

# 4. str.title()
text = "machine learning applications"
title_text = text.title()
print("Title:", title_text)

# 5. str.strip()
text = "  robotics research  "
stripped_text = text.strip()
print("Stripped:", f"'{stripped_text}'")

# 6. str.replace(old, new)
text = "robotic arm"
replaced_text = text.replace("arm", "hand")
print("Replaced:", replaced_text)
# 7. str.split(separator)
text = "robotics, AI, and automation"
split_text = text.split(", ")
print("Split:", split_text)

# 8. str.join(iterable)
words = ["AI", "in", "robotics"]
joined_text = " ".join(words)
print("Joined:", joined_text)

# 9. str.find(substring)
text = "artificial intelligence"
index = text.find("intelligence")
print("Find:", index)

# 10. str.format()
name = "robot"
version = 2
formatted_string = "The {} version is {}".format(name, version)
```

```python
print("Formatted:", formatted_string)

# 11. str.startswith(prefix)
text = "machine learning"
starts_with = text.startswith("machine")
print("Starts with 'machine':", starts_with)

# 12. str.endswith(suffix)
text = "robotics"
ends_with = text.endswith("ics")
print("Ends with 'ics':", ends_with)
```

```
Uppercase: ROBOTICS
Lowercase: artificial intelligence
Capitalized: Deep learning
Title: Machine Learning Applications
Stripped: 'robotics research'
Replaced: robotic hand
Split: ['robotics', 'AI', 'and automation']
Joined: AI in robotics
Find: 11
Formatted: The robot version is 2
Starts with 'machine': True
Ends with 'ics': True
```