1. Write a Python program to print Hello, World! to the standard output?

```
my_string = "Hello, World!"
print(my_string)
```

```
Hello, World!
```

2. Validate whether each given string is a 10-digit mobile number starting with 7, 8, or 9 using regular expressions?

A:- Some Code Explanation

- pattern = r'^[789]\d{9}$'
-

| Part | Meaning |
|---|---|
| `[789]` | The number **must start** with 7 or 8 or 9 |
| `\d` | Any digit (0–9) |
| `\d{9}` | Exactly **9 more digits** |
| **Total** | 1 digit (7/8/9) + 9 digits = **10 digits** |

```
import re # Regex method
n = int(input()) # Reads the first input → the number of strings you will check.Converts it into an integer.
pattern = r'^[789]\d{9}$'
for _ in range(n):
    number = input()
    if re.match(pattern, number):
        print("YES")
    else:
        print("NO")
```

```
2
9587456281
YES
1252478965
NO
```

3. "Given n name-email pairs in the format name user@email.com, print only those pairs where the email address is valid according to the rules: username starts with a letter and may contain letters, digits, _, -, or ., the domain contains only letters, and the extension is 1 to 3 letters long."?

```
import re
import email.utils
# Regex pattern for valid email
pattern = r'^[a-zA-Z][\w.-]*@[a-zA-Z]+\.[a-zA-Z]{1,3}$'
# Read number of inputs
n = int(input())
for _ in range(n):
    line = input().strip()
    # Parse the name and email
    name, addr = email.utils.parseaddr(line)
    # Validate the email using regex
    if re.fullmatch(pattern, addr):
        print(line)
```

```
2
DEXTER <dexter@hotmail.com>
DEXTER <dexter@hotmail.com>
VIRUS <virus!@variable.:p>
```

4. Given N lines of CSS code, print all valid HEX color codes (3 or 6 hexadecimal digits after '#', excluding selector names) in the order they appear?

```
import re

n = int(input())
inside = False

for _ in range(n):
    line = input()

    # If block starts after this line, check hex AFTER this line only
    if '{' in line:
        inside = True
        continue

    if '}' in line:
        inside = False

    # Search for HEX codes ONLY inside the block
    if inside:
        matches = re.findall(r'#[0-9A-Fa-f]{3,6}', line)
        for m in matches:
            if len(m) == 4 or len(m) == 7:   # #RGB or #RRGGBB
                print(m)
```

```
11
#BED
{
color: #FfFdF8; background-color:#aef;
#FfFdF8
#aef
font-size: 123px;

}
#Cab
{
background-color: #ABC;
#ABC
border: 2px dashed #fff;
#fff
}
```

5. **"Given N lines of HTML code, print all start tags, end tags, empty tags, and their attributes (with values or None) in the order they appear, ignoring anything inside HTML comments."**

---

**"First line contains N, followed by N lines of HTML code."**

Example (single line representation):

```
Input: 2 lines → <html><head><title>HTML Parser - I</title></head>  and  <body data-modal-target class='1'><h1>HackerRank</h1>
```

---

**"Print Start : tag, End : tag, Empty : tag, and ⟶ attribute > value (or None) exactly in order."**

Example Output (single-line representation):

```
Start: html, Start: head, Start: title, End: title, End: head, Start: body, -> data-modal-target > None, -> class > 1, Start:
```

```
from html.parser import HTMLParser
class MyHTMLParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print(f"Start : {tag}")
        for attr, value in attrs:
            print(f"-> {attr} > {value}")
    def handle_endtag(self, tag):
        print(f"End   : {tag}")
    def handle_startendtag(self, tag, attrs):
        print(f"Empty : {tag}")
        for attr, value in attrs:
            print(f"-> {attr} > {value}")
# Put your HTML here ↓
html_data = """
<html>
<head>
<title>Test</title>
</head>
<body>
<div data-modal-target>
<h1 class="1">Hello</h1>
<br/>
</body>
</html>
"""

parser = MyHTMLParser()
parser.feed(html_data)
```

```
Start : html
Start : head
Start : title
End   : title
End   : head
Start : body
Start : div
-> data-modal-target > None
Start : h1
```

```
-> class > 1
End   : h1
Empty : br
End   : body
End   : html
```

6. Write a program to parse HTML and print single-line comments, multi-line comments, and data (excluding newline data) using HTMLParser?

```python
from html.parser import HTMLParser

class MyHTMLParser(HTMLParser):

    def handle_comment(self, data):
        lines = data.split('\n')

        if len(lines) > 1:
            print(">>> Multi-line Comment")
            for line in lines:
                print(line)
        else:
            print(">>> Single-line Comment")
            print(data)

    def handle_data(self, data):
        if data.strip():     # ignore only newline
            print(">>> Data")
            print(data)

# Put HTML Here ⬇⬇⬇
html_data = """<!--[if IE 9]>IE9-specific content
<![endif]-->
<div> Welcome to HackerRank</div>
<!--[if IE 9]>IE9-specific content<![endif]-->"""

parser = MyHTMLParser()
parser.feed(html_data)
parser.close()
```

```
>>> Multi-line Comment
[if IE 9]>IE9-specific content
<![endif]
>>> Data
 Welcome to HackerRank
>>> Single-line Comment
[if IE 9]>IE9-specific content<![endif]
```

7. Write a program to parse HTML and print all tags, attributes, and attribute values (ignoring anything inside HTML comments)?

```python
from html.parser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print(tag)
        for name, value in attrs:
            print(f"-> {name} > {value}")

    def handle_startendtag(self, tag, attrs):
        print(tag)
        for name, value in attrs:
            print(f"-> {name} > {value}")

html = """
<head>
<title>HTML</title>
</head>
<object type="application/x-flash"
        data="your-file.swf"
        width="0" height="0">
<!-- <param name="movie" value="your-file.swf" /> -->
<param name="quality" value="high"/>
</object>
"""

parser = MyHTMLParser()
parser.feed(html)
parser.close()
```

```
head
title
object
```

```
-> type > application/x-flash
-> data > your-file.swf
-> width > 0
-> height > 0
param
-> name > quality
-> value > high
```