String Formatting

Using *f-strings*

Example 1 – No Real Formatting

```
first, last = 'Mike', 'Smith'
age = 26
message = f'{first} {last} is {age} years old.'
```

Result

Mike Smith is 26 years old.

Example 2 – Field Widths

```
first, last = 'Mike', 'Smith'
age = 26
message = f'{first:8} {last:3} is {age:10} years old.'
```

s for string 8 wide

s for string 3 wide

d for Dec 10 wide

Result

```
Mike     Smith is         26 years old.
```

Strings justify left

Numbers justify right

Example 3 – Changing Justification

```
first, last = 'Mike', 'Smith'
age = 26
message = f'{first:>8} {last:^12} is {age:<10} years old.'
```

Result

```
    Mike     Smith      is 26          years old.
```

Example 4 – Construct The Format String Dynamically

```
for places in range (5,21,5):
    message = f'pi to {places-1:2} places is {math.pi:0.{places}}'
    print(message)
```

- Result

```
pi to  4 places is 3.1416
pi to  9 places is 3.141592654
pi to 14 places is 3.14159265358979
pi to 19 places is 3.1415926535897931116
```

Example 5 – Add Commas to a Number

```
largeInt = 12345678901234567890
largeFloat = 123456789.123456789
message = f'{largeInt:,}\n{largeFloat:,}'
print(message)
```

, adds commas

- Result

```
1,234,567,890,123,456,789

123,456,789.123467
```

Example 6 – Numbers with No Formatting

```
for n in [-100, -10, 1, 10, 100]:
    message = f'n = {n}'
    print(message)
```

- Result

```
n = -100
n = -10
n = 1
n = 10
n = 100
```

Example 7 – Force A Plus (+) Sign

```
for n in [-100, -10, 1, 10, 100]:
    message = f'n = {n:+}'
    print(message)
```

- Result

```
n = -100
n = -10
n = +1
n = +10
n = +100
```

Example 8 – Right Justify

```
for n in [-100, -10, 1, 10, 100]:
    message = f'n = {n:>12}'
    print(message)
```

- Result

```
n =         -100
n =          -10
n =            1
n =           10
n =          100
```

Example 9 – Right Justify with Forced Plus (+) Sign

```
for n in [-100, -10, 1, 10, 100]:
    message = f'n = {n:>+12}'
    print(message)
```

- Result

```
n =         -100
n =          -10
n =           +1
n =          +10
n =         +100
```

Example 10 – Fill With A Different Character

```
for n in [-100, -10, 1, 10, 100]:
    message = f'n = {n:x>12}'
    print(message)
```

- Result

```
n = xxxxxxxx-100
n = xxxxxxxxx-10
n = xxxxxxxxxxx1
n = xxxxxxxxxx10
n = xxxxxxxxx100
```

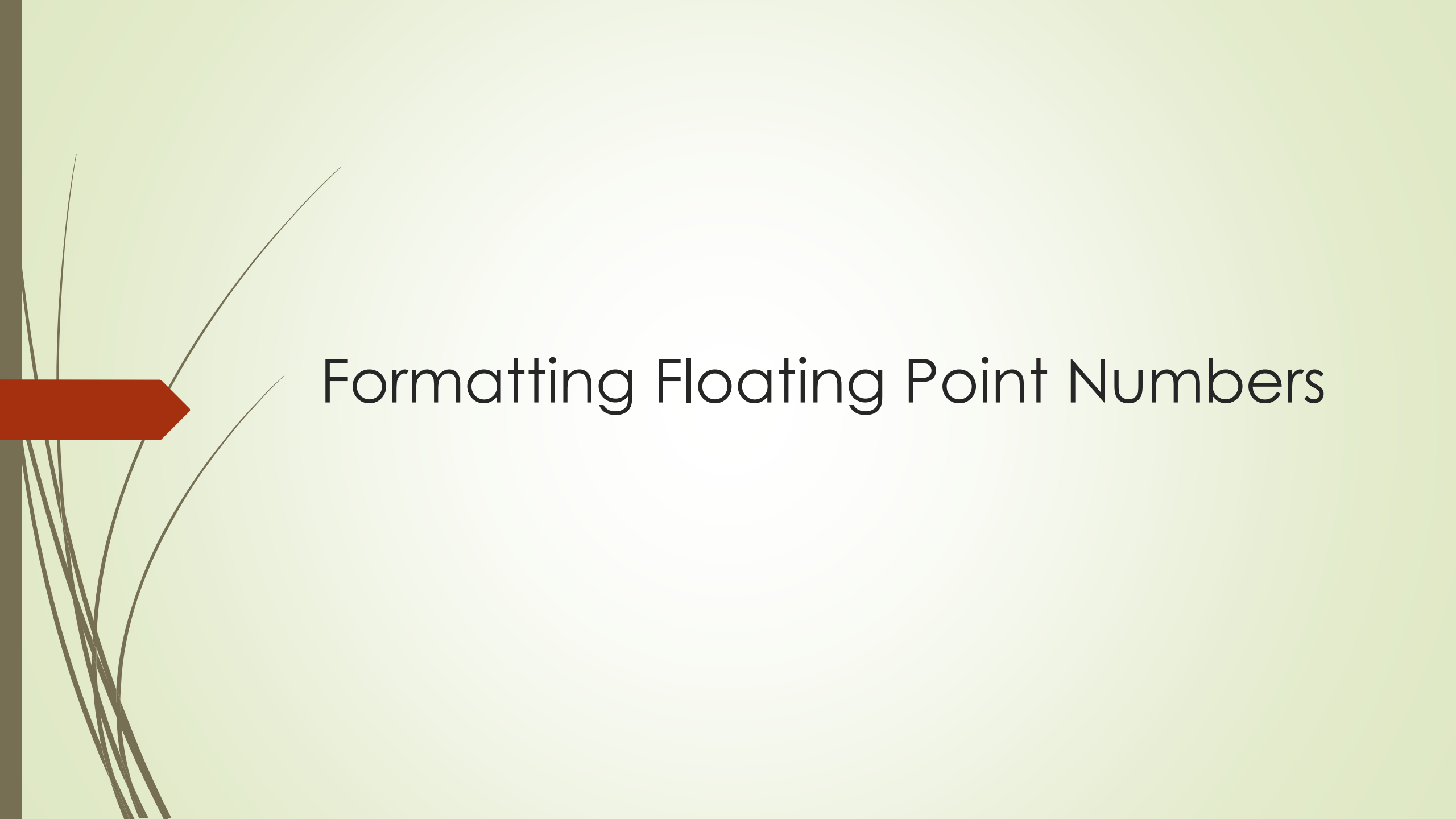
Example 11 – 2D List with Aligned Columns

```
data = [[8792,1,-109,112], [0,9383902,98, 87], [2881, 91, 837, -989]]
maxDataWidth = 0
for row in data:
    for item in row:
        if len(str(item)) > maxDataWidth:
            maxDataWidth = len(str(item))

pad = 2
for row in data:
    message = ''
    for item in row:
        message += f'{item:{maxDataWidth + pad}}'
    print(message)
```

- Result

```
  8792        1     -109      112
     0  9383902       98       87
  2881       91      837     -989
```


Formatting Floating Point Numbers

Example 12 – Field Width of 7 and 2 Decimal Places

```
n = 11.11111
s = f'X{n:7.2f}X'
```

- Result

```
X  11.11X
```

Example 13 – Width Unspecified and 2 Decimal Places

```
n = 11.11111
s = f'X{n:.2f}X
```

- Result

```
X11.11X
```

Example 16 – Use a Variable for Width

```
n = 11.11111
width = 9
s = f'X{n:{width}.2f}X'
```

- Result

```
X    11.11X
```

Example 17 – Use Variables for Both Width and Number of Places

```
n = 11.11111
width = 9
places = 3
s = f'X{n:{width}.{places}}X'
```

- Result

```
X   11.111X
```