Practice Problems and String Format

String → Practice Problems

1 Write a program to delete a given word in a String?

Requirements:

Input :	Output
Enter a string → “Hello Python Developers World”	Hello Python Developers
Enter a string to delete → “World”	

Program - 2

Write a Program to sort words in alphabetical order.

Input :

String = “good work people”

Output

```
good

people

work
```

Program 3

Write a program to count the characters in a String.

Input:

String : “Hello”

Output:

```
H count is 1
e  count is 1
l count is 2
O  count is 1
```

Program 4

Write a program to convert decimal to binary using string format .

What does end =' do in Python?

The end parameter in the print function is used to add any string. At the end of the output of the print statement in python.

By default, the print function ends with a newline.

Passing the whitespace to the end parameter (end=' ') indicates that the end character has to be identified by whitespace and not a newline.

For example:

```
print(“Toppr”, end=’ ‘)

print(“is awesome”)
```

Output:

```
Toppr is awesome
```

For example:

```
print(“toppr”, end=’ says ‘)

print(“you are awesome”)
```

Output:

```
toppr says you are awesome
```

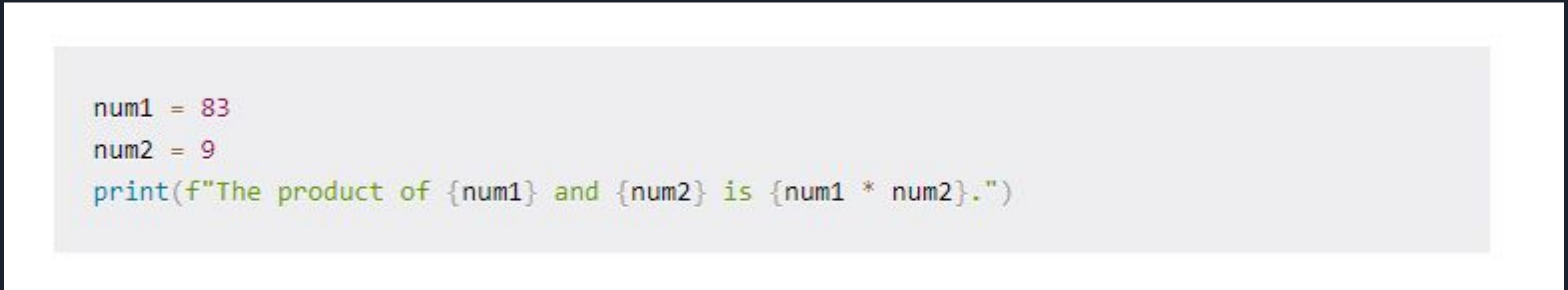
format() method in String

The `format()` method that is available with the string object is very versatile and powerful in formatting strings. Format strings contain curly braces `{}` as placeholders or replacement fields which get replaced.

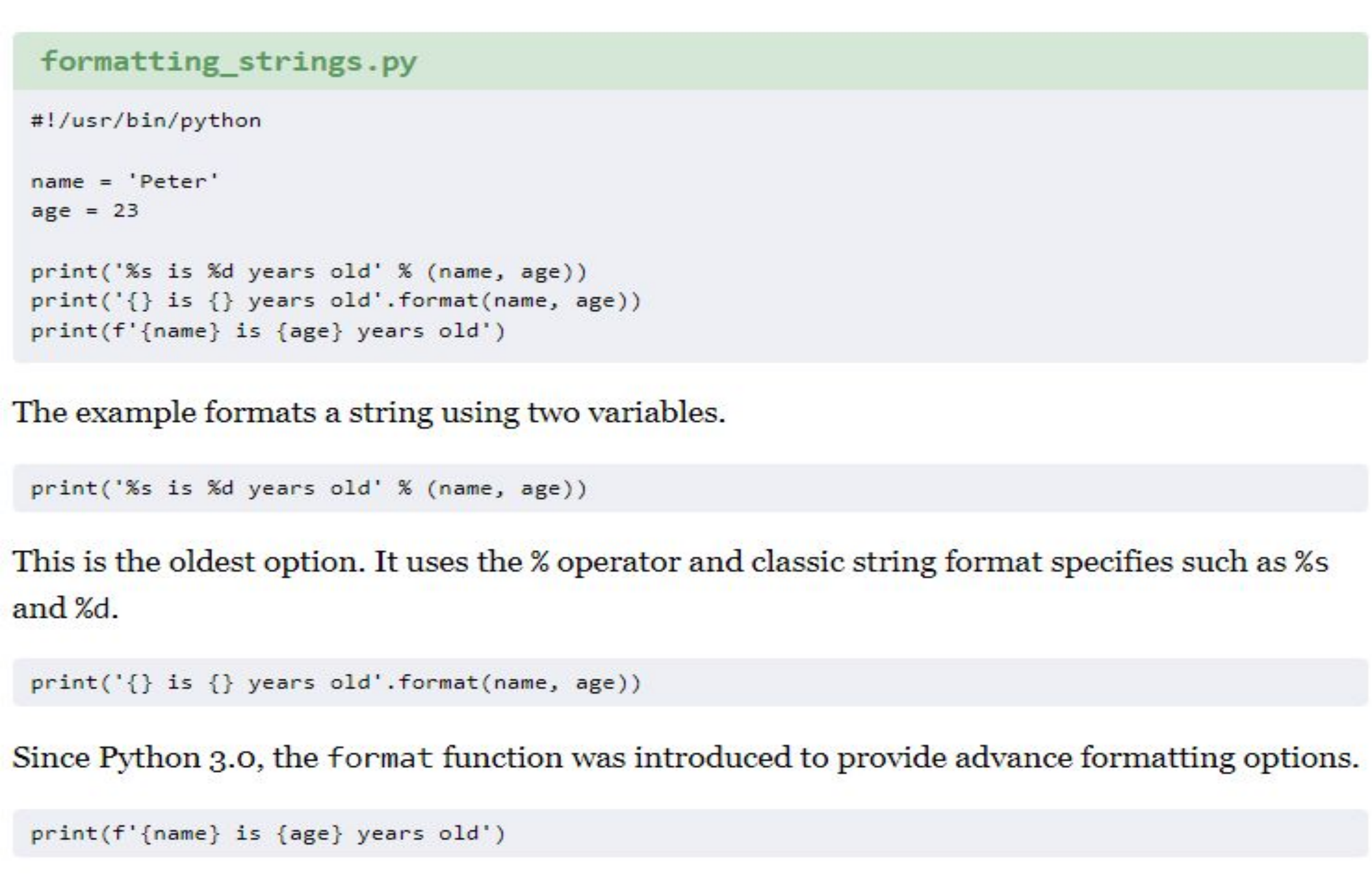
Example of String format

```
# Old Style formatting

# This prints out "John is 23 years old."
name = "John"
age = 23
print("%s is %d years old." % (name, age))
```

```
# The latest style formatting string using ( f” “)
```



```
num1 = 83
num2 = 9
print(f"The product of {num1} and {num2} is {num1 * num2}.")
```



formatting_strings.py

```
#!/usr/bin/python

name = 'Peter'
age = 23

print('%s is %d years old' % (name, age))
print('{} is {} years old'.format(name, age))
print(f'{name} is {age} years old')
```

The example formats a string using two variables.

```
print('%s is %d years old' % (name, age))
```

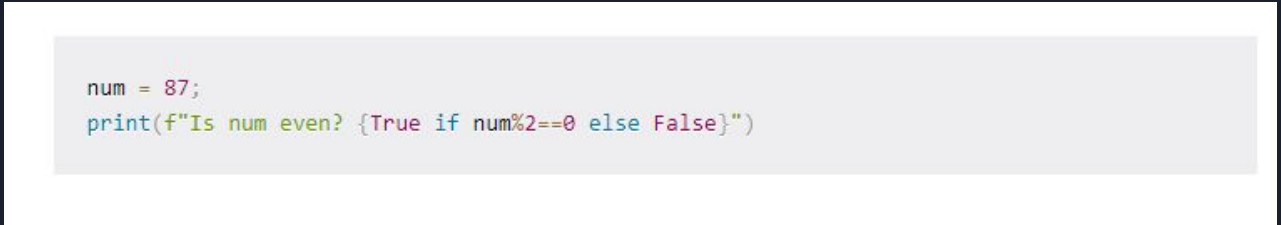
This is the oldest option. It uses the % operator and classic string format specifies such as %s and %d.

```
print('{} is {} years old'.format(name, age))
```

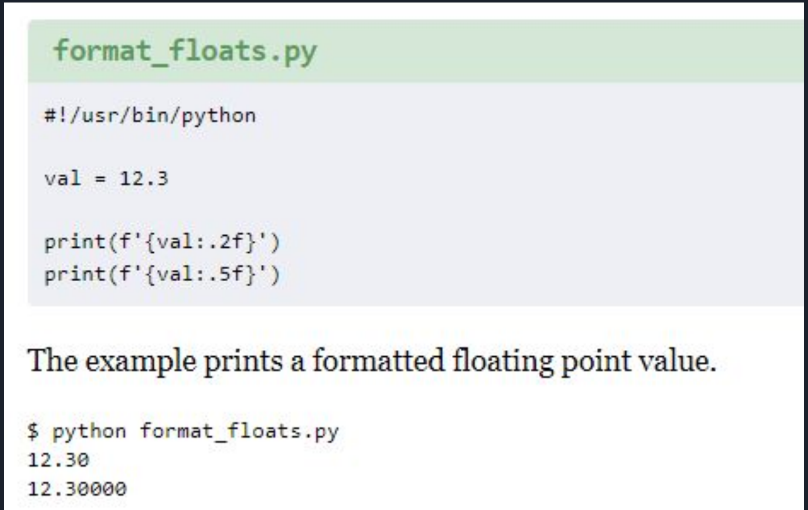
Since Python 3.0, the `format` function was introduced to provide advance formatting options.

```
print(f'{name} is {age} years old')
```

Examples



```
num = 87;
print(f"Is num even? {True if num%2==0 else False}")
```

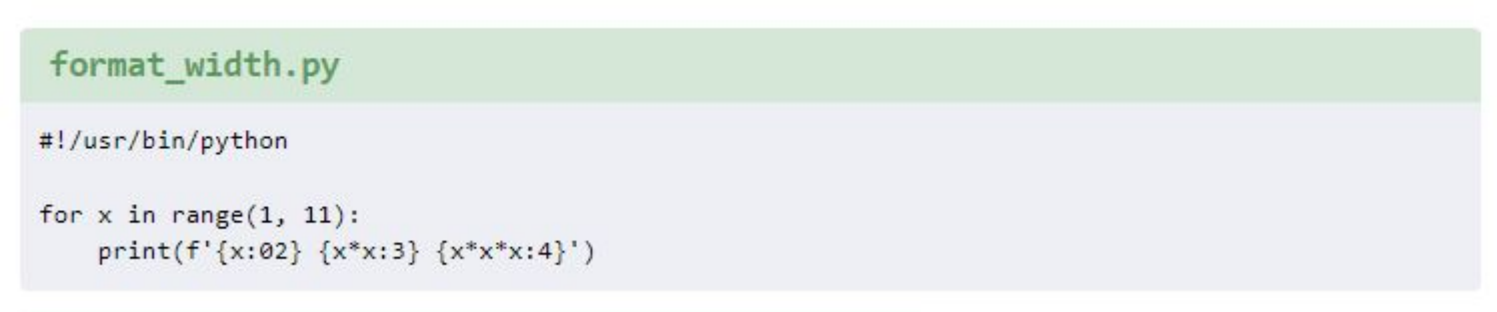


format_floats.py

```
#!/usr/bin/python

val = 12.3

print(f'{val:.2f}')
print(f'{val:.5f}')
```

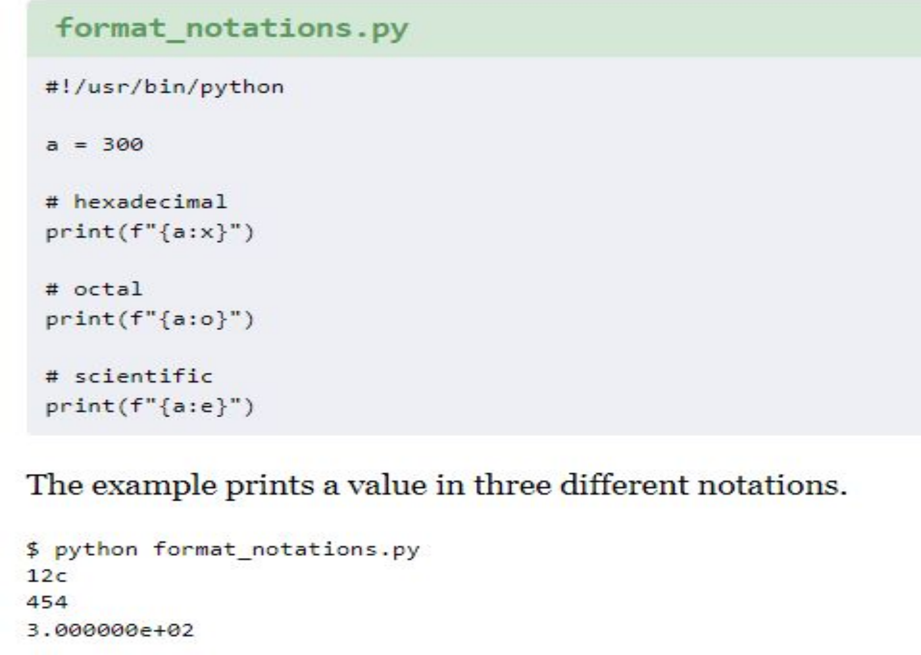
The example prints a formatted floating point value.

```
$ python format_floats.py
12.30
12.30000
```



format_width.py

```
#!/usr/bin/python

for x in range(1, 11):
    print(f'{x:02} {x*x:3} {x*x*x:4}')
```

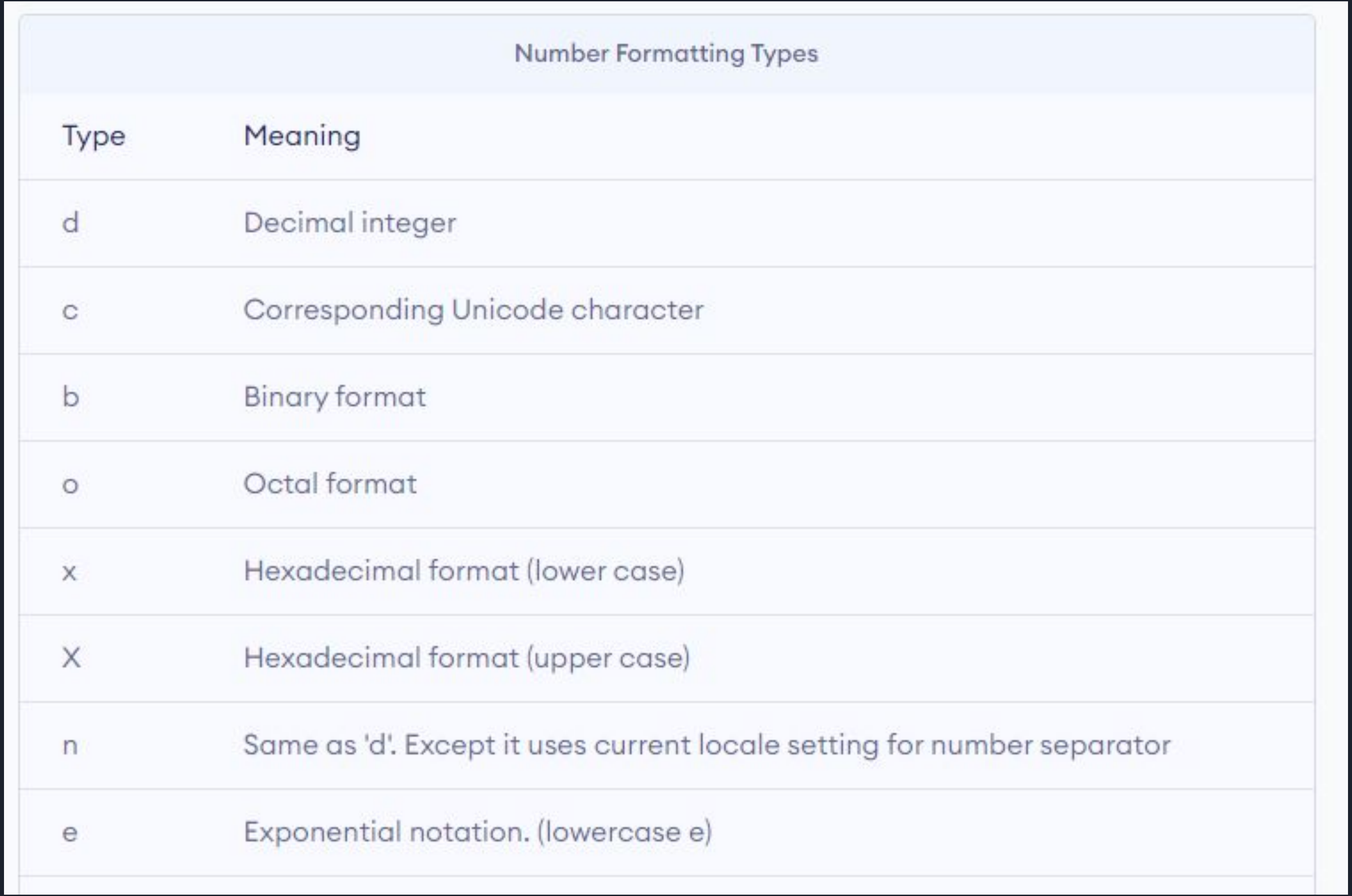


format_notations.py

```
#!/usr/bin/python

a = 300

# hexadecimal
print(f"{a:x}")

# octal
print(f"{a:o}")

# scientific
print(f"{a:e}")
```

The example prints a value in three different notations.

```
$ python format_notations.py
12c
454
3.000000e+02
```

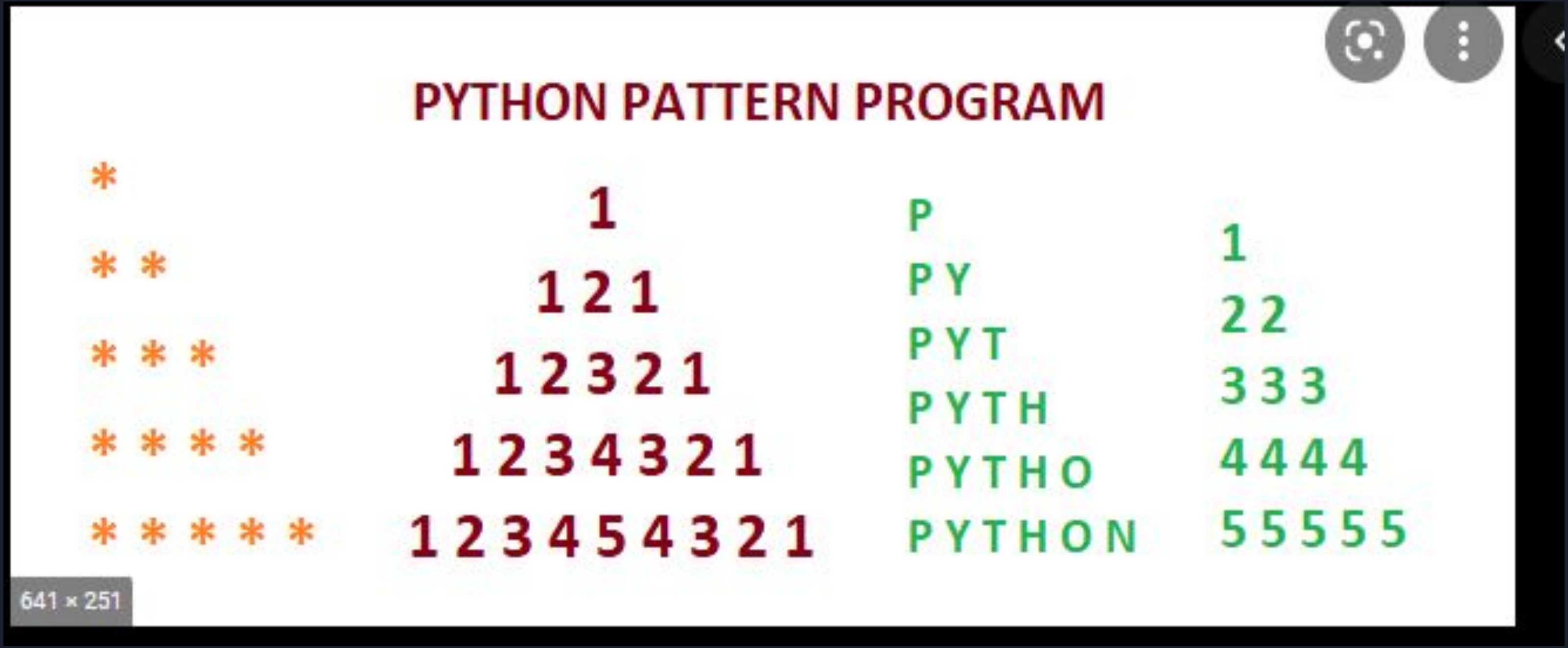


Number Formatting Types

Type	Meaning
d	Decimal integer
c	Corresponding Unicode character
b	Binary format
o	Octal format
x	Hexadecimal format (lower case)
X	Hexadecimal format (upper case)
n	Same as 'd'. Except it uses current locale setting for number separator
e	Exponential notation. (lowercase e)

Python Program to Find the Sum of Natural Numbers

Python Program to Find the Factors of a Number



PYTHON PATTERN PROGRAM

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

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

```
P
P Y
P Y T
P Y T H
P Y T H O
P Y T H O N
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```