Python

Table of contents

1 Python

```python
print("Hello, Python!")
```

```
Hello, Python!
```

Python

2 Python - Python

```python
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

3

Python -

```
#
name = "Alice"          #
age = 25                #
height = 5.6            #
is_student = True       #

print(f" : {name}")
print(f" : {age}")
print(f" : {height}")
print(f" : {is_student}")
```

```
 : Alice
 : 25
 : 5.6
 : True
```

3.1

```
#
first_name = "John"
last_name = "Doe"
age_2024 = 30
_private_var = "hidden"

#        :
# 2name = "invalid"     #
# first-name = "invalid" #
# class = "invalid"     #

print(f"   : {first_name} {last_name}")
```

: John Doe

4

```python
#
print("Hello")  #

"""



"""

def greet(name):
    """


    """
    return f"Hello, {name}!"

print(greet("Python"))
```

```
Hello
Hello, Python!
```

5

5.1

```
#
count = 42
negative = -17

#
price = 19.99
scientific = 1.5e-4  # 0.00015

#
complex_num = 3 + 4j

print(f" : {count}")
print(f"    : {price}")
print(f"   : {scientific}")
print(f"  : {complex_num}")
```

```
 : 42
    : 19.99
   : 0.00015
  : (3+4j)
```

5.2

```
#
single_quotes = 'Hello'
double_quotes = "World"
triple_quotes = """
    """
```

```
#
greeting = single_quotes + " " + double_quotes
print(greeting)

#
name = "Python"
version = 3.12

# f-strings
message = f"Welcome to {name} {version}!"
print(message)

# format()
message2 = "Welcome to {} {}!".format(name, version)
print(message2)

# %
message3 = "Welcome to %s %.1f!" % (name, version)
print(message3)
```

```
Hello World
Welcome to Python 3.12!
Welcome to Python 3.12!
Welcome to Python 3.1!
```

5.3

```
text = "Python Programming"

#
print(f" : {len(text)}")

#
print(f"  : {text.upper()}")
print(f"  : {text.lower()}")
print(f"    : {text.title()}")

#
print(f"'Python'  : {text.startswith('Python')}")
print(f"'gram'  : {'gram' in text}")
print(f" : {text.replace('Python', 'Java')}")

#
print(f"  6 : {text[:6]}")
print(f"  11 : {text[7:]}")
print(f"2  : {text[::2]}")
```

```
 : 18
  : PYTHON PROGRAMMING
  : python programming
    : Python Programming
'Python'   : True
'gram'  : True
 : Java Programming
  6  : Python
  11 : Programming
2   : Pto rgamn
```

5.4

```
#
is_python_fun = True
is_difficult = False

#
print(f"AND: {is_python_fun and is_difficult}")
print(f"OR: {is_python_fun or is_difficult}")
print(f"NOT: {not is_difficult}")

#    - True/False
print(f"   : {bool('')}")        # False
print(f"    : {bool('hello')}") # True
print(f" : {bool(0)}")                    # False
print(f"  : {bool(42)}")              # True
print(f"   : {bool([])}")           # False
print(f"    : {bool([1, 2])}")   # True
```

```
AND: False
```

```
OR: True
NOT: True
  : False
   : True
 : False
  : True
  : False
   : True
```

6

6.1

```python
a, b = 10, 3

print(f" : {a} + {b} = {a + b}")
print(f" : {a} - {b} = {a - b}")
print(f" : {a} * {b} = {a * b}")
print(f" : {a} / {b} = {a / b}")
print(f"    : {a} // {b} = {a // b}")
print(f" : {a} % {b} = {a % b}")
print(f"  : {a} ** {b} = {a ** b}")
```

```
 : 10 + 3 = 13
 : 10 - 3 = 7
 : 10 * 3 = 30
 : 10 / 3 = 3.3333333333333335
    : 10 // 3 = 3
 : 10 % 3 = 1
  : 10 ** 3 = 1000
```

6.2

```python
x, y = 5, 10

print(f"{x} == {y}: {x == y}")   #
print(f"{x} != {y}: {x != y}")   #
print(f"{x} < {y}: {x < y}")     #
print(f"{x} > {y}: {x > y}")     #
print(f"{x} <= {y}: {x <= y}")   #
print(f"{x} >= {y}: {x >= y}")   #
```

```
5 == 10: False
5 != 10: True
5 < 10: True
5 > 10: False
5 <= 10: True
5 >= 10: False
```

6.3

```
num = 10
print(f"  : {num}")

num += 5    # num = num + 5
print(f"+=5  : {num}")

num -= 3    # num = num - 3
print(f"-=3  : {num}")

num *= 2    # num = num * 2
print(f"*=2  : {num}")

num //= 3   # num = num // 3
print(f"//=3  : {num}")
```

```
  : 10
+=5  : 15
-=3  : 12
*=2  : 24
//=3  : 8
```

7

7.1

```
#  : input()
# name = input("    ")
# print(f"   {name}!")

#
# age_str = input("    ")
# age = int(age_str)
# print(f"  {age}  ")

#
# age = int(input("    "))

#
name = "Alice"  #
age = 25        #

print(f"   {name}!")
print(f"  {age}  ")
```

```
   Alice!
  25
```

7.2

```
import math

#    print
print("   ")
```

```python
#
print("  ", " ", "  ", "print ")

#
print("A", "B", "C", sep="-")
print("   ", end=" ")
print("   ")

#
pi = math.pi
print(f"  : {pi}")
print(f"    2 : {pi:.2f}")
print(f"     : {pi:.2e}")

#
number = 42
print(f"  : {number:>10}")
print(f"  : {number:<10}")
print(f"   : {number:^10}")
print(f"    : {number:05}")
```

```
       print
A-B-C

  : 3.141592653589793
    2 : 3.14
     : 3.14e+00
  :         42
  : 42
   :     42
    : 00042
```

8

Python

```
#
age = 18
if age >= 18:
    print("     ")
    print("    ")
else:
    print("      ")
    print("      ")

print("  if    ")
```

if

8.1

```
#    -        IndentationError
if True:
    print("  4      ")
        print("  8      ")  #

#    -      IndentationError
if True:
print("       ")  #

#    -
if True:
    print("          ")
    print("    ")
```

9

9.1 1:

```python
#
first_name = "John"
last_name = "Doe"
age = 30
city = "New York"
hobby = "programming"

#
print("===    ===")
print(f" : {first_name} {last_name}")
print(f" : {age} ")
print(f" : {city}")
print(f" : {hobby}")
print("="*30)
```

```
===    ===
 : John Doe
 : 30
 : New York
 : programming
==============================
```

9.2 2:

```
#
num1 = 15
num2 = 4

print(f" 1: {num1}")
print(f" 2: {num2}")
print("-" * 20)
print(f" : {num1} + {num2} = {num1 + num2}")
print(f" : {num1} - {num2} = {num1 - num2}")
print(f" : {num1} × {num2} = {num1 * num2}")
print(f" : {num1} ÷ {num2} = {num1 / num2:.2f}")
print(f"  : {num1}^{num2} = {num1 ** num2}")
```

```
 1: 15
 2: 4
--------------------
 : 15 + 4 = 19
 : 15 - 4 = 11
 : 15 × 4 = 60
 : 15 ÷ 4 = 3.75
  : 15^4 = 50625
```

9.3 3:

```
#
sentence = "python is an amazing programming language"

print(f"   : {sentence}")
print(f" : {len(sentence)}  ")
print(f"  : {sentence.count(' ') + 1}")
print(f"    : {sentence.title()}")
print(f"  : {sentence.upper()}")
print(f"   : {sentence.capitalize()}")
print(f"'python' 'Python'  : {sentence.replace('python', 'Python')}")
```

```
   : python is an amazing programming language
 : 41
  : 6
```

```
    : Python Is An Amazing Programming Language
  : PYTHON IS AN AMAZING PROGRAMMING LANGUAGE
   : Python is an amazing programming language
'python' 'Python'  : Python is an amazing programming language
```

10

1. - 4
2. -
3. **vs** - input()
4. - `Name` `name`
5. - Unicode

11

1. :

   ```
   #
   x = 25

   #
   student_age = 25
   ```

2. **PEP 8** :

   ```
   #    : snake_case
   user_name = "Alice"

   #   : UPPER_CASE
   MAX_ATTEMPTS = 3

   #   : PascalCase
   # UserAccount
   ```

3. **f-strings** Python 3.6+

4.

5. **80**

12

1. `print(type(5))`
2. `'hello'` `"hello"`
3.
4.
5. `10 // 3`

12.1

1. `<class 'int'>`
2. -
3. TypeError -
4. : `"""` `'''`
5. `3`

13

Python