PRODDEC PYTHON DAY 1

print() function

In [ ]:
```
# print a content to the output screen.
```

Printing Numbers

In [2]:
```
print(114541)
print(115151.21151)
```
```
114541
115151.21151
```

Addition

In [3]:
```
print(15454 + 5151)
print(11155 + 515125.2415)
```
```
20605
526280.2415
```

Subtraction

In [4]:
```
print(111156 - 515152151)
print(15241.255 - 4584.11)
```
```
-515040995
10657.145
```

Multiplication

In [5]:
```
print(14515 * 51515)
print(5211 * 521)
```
```
747740225
2714931
```

Division

Actual Division

In [6]:
```
print(1515/2)
```
```
757.5
```

Floor Division (Output will be integer.)

In [7]:
```
print(5//2)
```
```
2
```

Zero Division Error

In [8]:
```
print(42/0)
```
```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-8-bb2887efa86d> in <module>
----> 1 print(42/0)

ZeroDivisionError: division by zero
```

Getting the remainder on division (modulo operator)

In [9]:
```
print(7%3)
print(12%5)
```
```
1
2
```

Getting the Power

Using the **

In [1]:
```
print(2 ** 3)
print(5 ** 2)
print(3 ** 4)
```
```
8
25
81
```

Using the pow() function

In [11]:
```
print(pow(2,3))
print(pow(5,2))
print(pow(3,4))
```
```
8
25
81
```

Getting the root of a number

In [12]:
```
print(2 ** 0.5)
print(3 ** 0.5)
print(36 ** 0.5)

# Using the pow() function

print(pow(2,0.5))
print(pow(3,0.5))
print(pow(36,0.5))
```
```
1.4142135623730951
1.7320508075688772
6.0
1.4142135623730951
1.7320508075688772
6.0
```

Getting the remainder when a^b is divided by c.

In [2]:
```
# print(pow(a,b,c))
print(pow(2,5,6))
print(pow(2,3,7))
```
```
2
1
```

Strings

In [ ]:
```
"""
A string is a sequence of characters.
A character is simply a symbol.
For example, the English language has 26 characters.
Computers do not deal with characters, they deal with numbers (binary).
Even though you may see characters on your screen,
internally it is stored and manipulated as a combination of 0s and 1s.

This conversion of character to a number is called encoding,
and the reverse process is decoding.
ASCII and Unicode are some of the popular encodings used.

In Python, a string is a sequence of Unicode characters.
Unicode was introduced to include every character in all
languages and bring uniformity in encoding.
"""
```

Creating strings in python

In [ ]:
```
"""
Strings can be created by enclosing characters inside a
single quote or double-quotes. Even triple quotes can be
used in Python but generally used to represent
multiline strings and docstrings.
"""
```

Printing the string

In [4]:
```
print('dog')
print("cat")
print('''giraffe''')
print("""cow""")
print('255145151515415152')
print("5154151")
print("""66565""")
print('''54154154''')
print('51515.21')
```
```
dog
cat
giraffe
cow
255145151515415152
5154151
66565
54154154
51515.21
```

Printing the ' in the string

In [15]:
```
print("Dany's dog")
print('''Peter's watch''')
print("""Sucy's cat""")
```
```
Dany's dog
Peter's watch
Sucy's cat
```

Printing the " in the string

In [16]:
```
print('Dany"s dog')
print('''Peter"s watch''')
print("""Sucy"s cat""")
```
```
Dany"s dog
Peter"s watch
Sucy"s cat
```

Printing the ' and " in string

In [17]:
```
print('''Peter's dog and Sucy"s cat''')
print("""Peter's dog and Sucy"s cat""")
```
```
Peter's dog and Sucy"s cat
Peter's dog and Sucy"s cat
```

Doing two commands at a time in python shell.

In [18]:
```
print("Hello");print("Goodbye!")
```
```
Hello
Goodbye!
```

Arguments

In [ ]:
```
# The values that we passed into a function.
```

Adding multiple arguments in print function.

In [ ]:
```
# The default separation is space.
```

In [19]:
```
print(1,2,3,4)
print("dog","cat","vehicle",123,34354,4324234.34)
```
```
1 2 3 4
dog cat vehicle 123 34354 4324234.34
```

Adding a separation

In [20]:
```
print(23,224,35646,sep="*")
print("spam","eggs","cat",sep="-")
```
```
23*224*35646
spam-eggs-cat
```

EOL Error (End Of Line)

In [21]:
```
# This is due to absence of a single quotation mark.
print('Hello World)
```
```
  File "<ipython-input-21-635980887880>", line 1
    print('Hello World)
                       ^
SyntaxError: EOL while scanning string literal
```

In [22]:
```
# This is due to absence of a double quotation mark.
print("Hello World)
```
```
  File "<ipython-input-22-69ef97eb7c7b>", line 1
    print("Hello World)
                       ^
SyntaxError: EOL while scanning string literal
```

Getting the type of a value.

type() function

In [23]:
```
print(type(12121))
print(type(1554.541514))
print(type("Spam and eggs"))
```
```
<class 'int'>
<class 'float'>
<class 'str'>
```

Finding the length of a string.

len() function

In [24]:
```
print(len("fsdfsdfg sdgdsg"))
```
```
15
```

TypeError for integer number

In [25]:
```
# Integer doesn't have length.
print(len(11515))
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-25-0c37d20ec6fd> in <module>
----> 1 print(len(11515))

TypeError: object of type 'int' has no len()
```

max() function

In [5]:
```
# Find maximum from the values given.
print(max(232423,434335,345435,64556465645654,56))
print(max("spam","eggs","dog"))
print(max('a',"A",'B',"b"))
```
```
64556465645654
spam
b
```

min() function

In [27]:
```
# Finding the minimum values.
print(min(232423,434335,345435,64556465645654,56))
print(min("spam","eggs","dog"))
print(min('a',"A",'B',"b"))
```
```
56
dog
A
```

In [ ]:
```
"""
In case of string max() and min() work based on the
ASCII value of each character in the string.
ASCII of A->65  ASCII of a->97
"""
```

abs() function

Find absolute value of a number

In [28]:
```
print(abs(-212))
print(abs(2121))
print(abs(-1251.521))
print(abs(2121.212))
```
```
212
2121
1251.521
2121.212
```

bin() function

Finding the binary representation of a number

In [29]:
```
print(bin(12))
print(bin(99))
print(bin(-14))
```
```
0b1100
0b1100011
-0b1110
```

Escape characters

In [30]:
```
# TAB
print("Hello\tWorld")
print("Hello\t\tWorld")
```
```
Hello	World
Hello		World
```

In [31]:
```
# NEW LINE
print("Hello\nWorld")
```
```
Hello
World
```

In [6]:
```
print("Peter\'s Dog")
print('Peter\'s Dog')
print("Sucys\" Cat")
print('Sucys\" Cat')
print("The number 42\\0 cause zero division error")
```
```
Peter's Dog
Peter's Dog
Sucys" Cat
Sucys" Cat
The number 42\0 cause zero division error
```

int() function

Converting string into integer

In [7]:
```
print(int("121551541"))
print(int("99992125"))
```
```
121551541
99992125
```

str() function

convert integer or float to string

In [34]:
```
print(str(21561515))
print(str(144541.515))
```
```
21561515
144541.515
```

float() function

Convert string to a float number

In [35]:
```
print(float("5145.241541"))
```
```
5145.241541
```

In [ ]: