

Arithmetic Operators inside `print()`

This notebook covers the use of arithmetic operators inside a `print` statement. Combining numeric and string values together with an arithmetic operator inside a `print` statement leads to different results depending on the argument.

In [3]:

```
'''
When the arithmetic operator is the `+` operator, both operands need to be numeric values or both
should be string values.
The numeric value can be either `int` or `float`. If one of the operands is numeric and the other
is string, you will
get an error.
'''

x = 5
y = 10
z = 5.5
u = 'hello'
v = 'Hi'

print(x + y) # Valid
print(x + z) # Valid
print(u + v) # Valid
#print(x + u) # Invalid
print(z + u) # Invalid
```

```
15
10.5
helloHi
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-3-d7861cf1609d> in <module>()
      15 print(u + v) # Valid
      16 #print(x + u) # Invalid
--> 17 print(z + u) # Invalid
```

TypeError: unsupported operand type(s) for +: 'float' and 'str'

In [6]:

```
'''
When the arithmetic operator is the * operator, either both should be numeric values or one of the
operands
can be string while the other is int. Using the * operator with two string operands will result in
an error.
'''

int1 = 5
int2 = 10
float1 = 5.5
str1 = 'hello'
str2 = 'Hi'

print(int1 * int2) # Valid
print(int1 * float1) # Valid
print(str1 * int1) # Valid
#print(str1 * str2) # Invalid
#print(str1 * float1) # Invalid
```

```
50
27.5
hellohellohellohellohello
```