1.6_Solutions

January 23, 2023

1. Compute the product of three numbers (6, -4, 2) and print the result on screen

[1]:
```
x = 6 * (-4) * 2
print('the result is',x)
```

```
the result is -48
```

2. Use python to print the expression 'I like using 'quotes' and also "quotes" in my text.

[2]:
```
print("I like using 'quotes' and also", '"quotes" in my text')
```

```
I like using 'quotes' and also "quotes" in my text
```

[3]:
```
print("I like using 'quotes' and also " + '"quotes" in my text')
```

```
I like using 'quotes' and also "quotes" in my text
```

[4]:
```
print(" I like using 'quotes' and also \"quotes\" in my text")
```

```
 I like using 'quotes' and also "quotes" in my text
```

3. Print the result of the following operations using python:
    1. $5+4+3+2+1$
    2. $5*4*3*2*1$
    3. 6^4

[5]:
```
print(5+4+3+2+1)
```

```
15
```

[6]:
```
print(5*4*3*2*1)
```

```
120
```

[7]:
```
print(6**4)
```

```
1296
```

4. what is the rest of dividing 8764 by 7?

[8]:
```
r= 8764 % 7
print('the result is',r)
```

```
the result is 0
```

[9]:
```
8764 / 7
```

[9]:
```
1252.0
```

5. and the mod division of the previous numbers?

6. verify using a boolean expression in pytyhon that $4^3 < 3^4$

[10]:
```
x=(4**3 < 3**4)
print(x)
```

```
True
```

7. Print all the available functions in the library called **numba**

[11]:
```
import numba
```

[12]:
```
dir(numba)
```

[12]:
```
['ByteCodeSupportError',
 'CompilerError',
 'ConstantInferenceError',
 'DeprecationError',
 'ForbiddenConstruct',
 'ForceLiteralArg',
 'IRError',
 'InternalError',
 'InternalTargetMismatchError',
 'LiteralTypingError',
 'LoweringError',
 'NotDefinedError',
 'NumbaAssertionError',
 'NumbaAttributeError',
 'NumbaDebugInfoWarning',
 'NumbaDeprecationWarning',
 'NumbaError',
 'NumbaExperimentalFeatureWarning',
 'NumbaIRAssumptionWarning',
 'NumbaIndexError',
 'NumbaInvalidConfigWarning',
 'NumbaKeyError',
 'NumbaNotImplementedError',
 'NumbaParallelSafetyWarning',
 'NumbaPedanticWarning',
 'NumbaPendingDeprecationWarning',
 'NumbaPerformanceWarning',
 'NumbaRuntimeError',
 'NumbaTypeError',
 'NumbaTypeSafetyWarning',
```

```
'NumbaValueError',
'NumbaWarning',
'RedefinedError',
'RequireLiteralValue',
'TypingError',
'UnsupportedError',
'UnsupportedParforsError',
'UnsupportedRewriteError',
'UntypedAttributeError',
'VerificationError',
'__all__',
'__builtins__',
'__cached__',
'__doc__',
'__file__',
'__loader__',
'__name__',
'__package__',
'__path__',
'__spec__',
'__version__',
'_devicearray',
'_dispatcher',
'_dynfunc',
'_ensure_critical_deps',
'_ensure_llvm',
'_helperlib',
'_min_llvm_version',
'_min_llvmlite_version',
'_try_enable_svml',
'_version',
'b1',
'bool_',
'boolean',
'byte',
'c16',
'c8',
'carray',
'cfunc',
'char',
'cloudpickle',
'complex128',
'complex64',
'config',
'core',
'cpython',
'deferred_type',
```

```
'double',
'errors',
'experimental',
'extending',
'f4',
'f8',
'farray',
'ffi',
'ffi_forced_object',
'float32',
'float64',
'float_',
'from_dtype',
'gdb',
'gdb_breakpoint',
'gdb_init',
'generated_jit',
'get_num_threads',
'guvectorize',
'i1',
'i2',
'i4',
'i8',
'int16',
'int32',
'int64',
'int8',
'int_',
'intc',
'intp',
'jit',
'jit_module',
'literal_unroll',
'literally',
'llvmlite',
'long_',
'longlong',
'misc',
'njit',
'none',
'np',
'numba',
'objmode',
'optional',
'parfors',
'platform',
'pndindex',
```

```
 'prange',
 're',
 'set_num_threads',
 'short',
 'size_t',
 'ssize_t',
 'stencil',
 'stencils',
 'sys',
 'test',
 'threading_layer',
 'typed',
 'typeof',
 'types',
 'u1',
 'u2',
 'u4',
 'u8',
 'uchar',
 'uint',
 'uint16',
 'uint32',
 'uint64',
 'uint8',
 'uintc',
 'uintp',
 'ulong',
 'ulonglong',
 'ushort',
 'vectorize',
 'version_info',
 'void',
 'warnings']
```

8. Use the math package to compute the following expression $e^{\pi} - 1$

[13]:
```
import math
```

[14]:
```
math.pi
```

[14]:
```
3.141592653589793
```

[15]:
```
math.e
```

[15]:
```
2.718281828459045
```

[16]:
```
y = math.exp(math.pi)-1
print(y)
```

```
22.140692632779267
```

[17]:
```
print(round(y, 3))
```

```
22.141
```

[18]:
```
from math import exp, pi
```

[19]:
```
print(exp(pi)-1)
```

```
22.140692632779267
```

[20]:
```
import math as mmm
```

[21]:
```
mmm.exp(mmm.pi)-1
```

[21]:
```
22.140692632779267
```

[ ]:
```

```