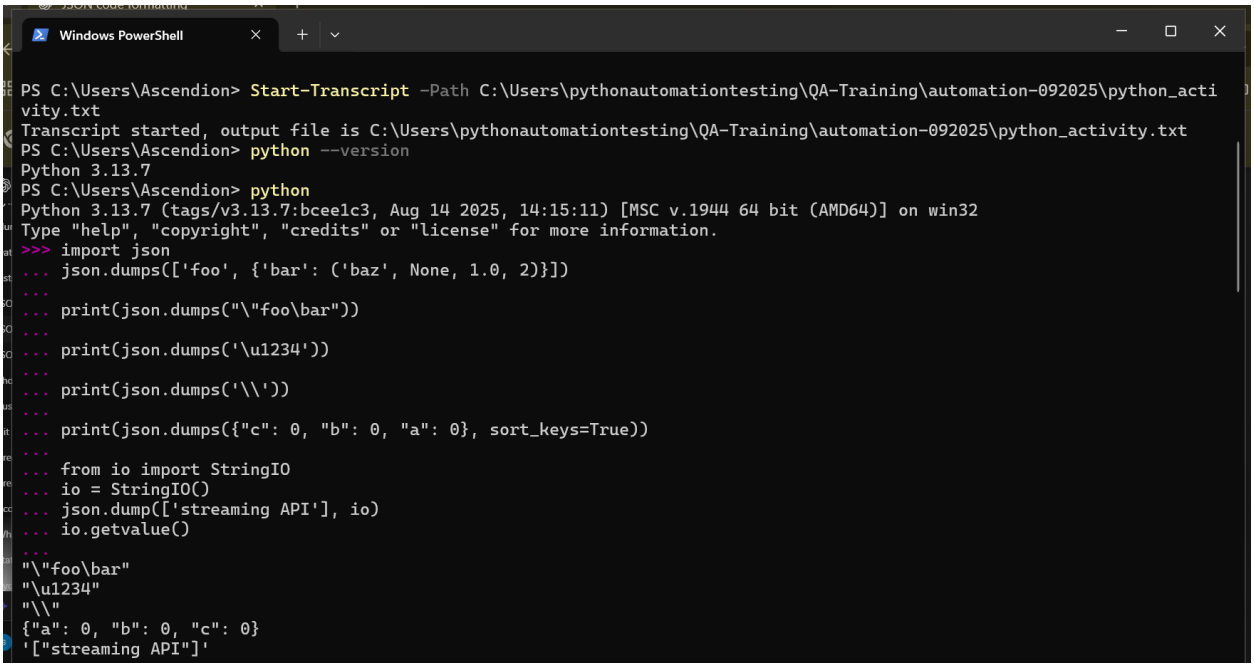
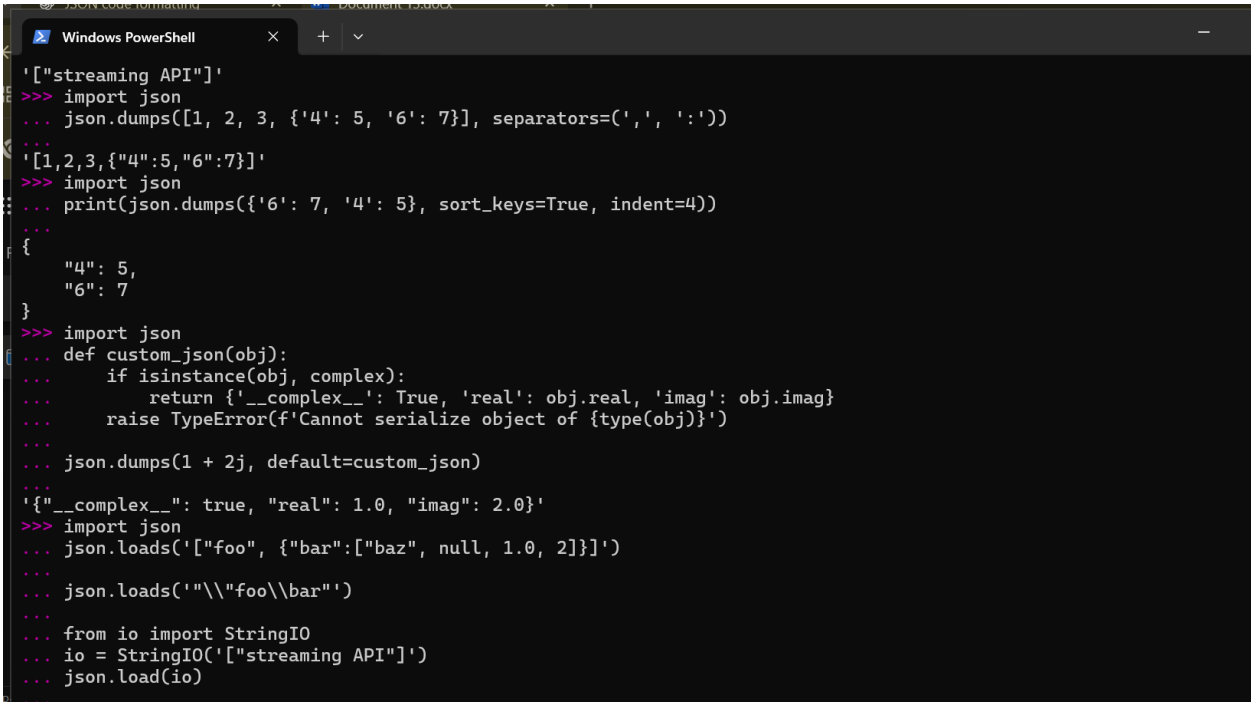
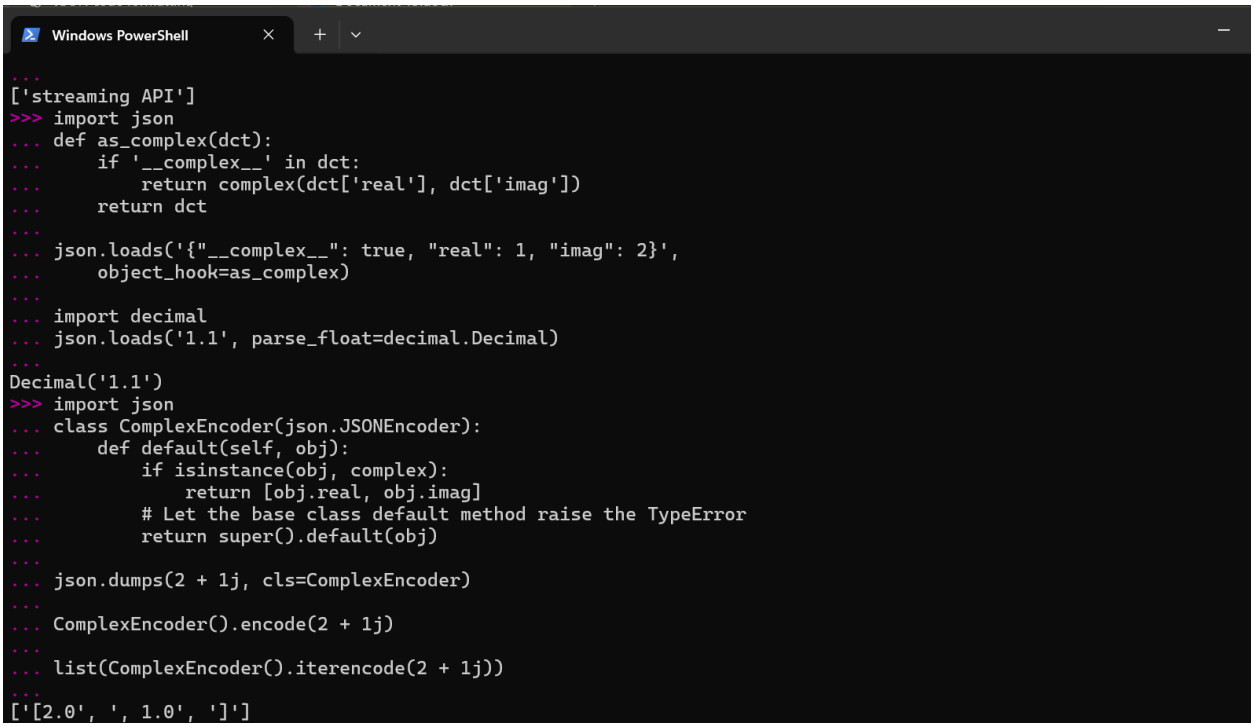
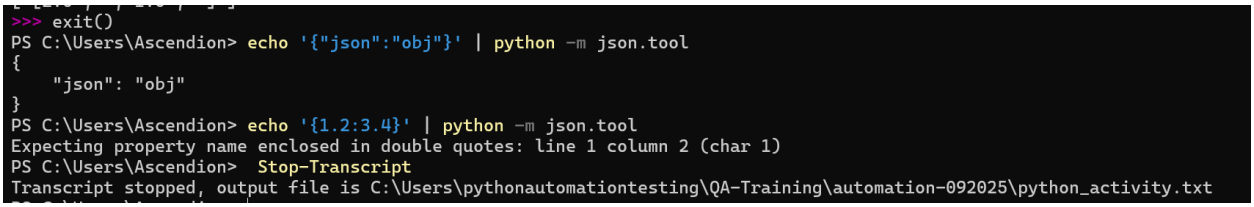

Windows PowerShell
PS C:\Users\Ascendion> Start-Transcript -Path C:\Users\pythonautomationtesting\QA-Training\automation-092025\python_activity.txt
Transcript started, output file is C:\Users\pythonautomationtesting\QA-Training\automation-092025\python_activity.txt
PS C:\Users\Ascendion> python --version
Python 3.13.7
PS C:\Users\Ascendion> python
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import json
... json.dumps(['foo', {'bar': ('baz', None, 1.0, 2)}])
...
... print(json.dumps("\"foo\bar"))
...
... print(json.dumps('\u1234'))
...
... print(json.dumps('\\'))
...
... print(json.dumps({"c": 0, "b": 0, "a": 0}, sort_keys=True))
...
... from io import StringIO
... io = StringIO()
... json.dump(['streaming API'], io)
... io.getvalue()
...
"\"foo\bar"
"\u1234"
"\\"
{"a": 0, "b": 0, "c": 0}
'["streaming API"]'


Windows PowerShell
'["streaming API"]'
>>> import json
... json.dumps([1, 2, 3, {'4': 5, '6': 7}], separators=(',', ':'))
...
'[1,2,3,{"4":5,"6":7}]'
>>> import json
... print(json.dumps({'6': 7, '4': 5}, sort_keys=True, indent=4))
...
{
    "4": 5,
    "6": 7
}
>>> import json
... def custom_json(obj):
...   if isinstance(obj, complex):
...   return {'__complex__': True, 'real': obj.real, 'imag': obj.imag}
...   raise TypeError(f'Cannot serialize object of {type(obj)}')
...
... json.dumps(1 + 2j, default=custom_json)
...
'{"__complex__": true, "real": 1.0, "imag": 2.0}'
>>> import json
... json.loads('["foo", {"bar":["baz", null, 1.0, 2]}]')
...
... json.loads('"\\"foo\\bar"')
...
... from io import StringIO
... io = StringIO('["streaming API"]')
... json.load(io)


Windows PowerShell
... ['streaming API']
>>> import json
... def as_complex(dct):
...   if '__complex__' in dct:
...   return complex(dct['real'], dct['imag'])
...   return dct
...
... json.loads('{"__complex__": true, "real": 1, "imag": 2}',
...   object_hook=as_complex)
...
... import decimal
... json.loads('1.1', parse_float=decimal.Decimal)
...
Decimal('1.1')
>>> import json
... class ComplexEncoder(json.JSONEncoder):
...   def default(self, obj):
...   if isinstance(obj, complex):
...   return [obj.real, obj.imag]
...   # Let the base class default method raise the TypeError
...   return super().default(obj)
...
... json.dumps(2 + 1j, cls=ComplexEncoder)
...
... ComplexEncoder().encode(2 + 1j)
...
... list(ComplexEncoder().iterencode(2 + 1j))
...
['[2.0', ', 1.0', ']']


>>> exit()
PS C:\Users\Ascendion> echo '{"json":"obj"}' | python -m json.tool
{
"json": "obj"
}
PS C:\Users\Ascendion> echo '{1.2:3.4}' | python -m json.tool
Expecting property name enclosed in double quotes: line 1 column 2 (char 1)
PS C:\Users\Ascendion> Stop-Transcript
Transcript stopped, output file is C:\Users\pythonautomationtesting\QA-Training\automation-092025\python_activity.txt