Python的Str方法以及repr方法返回的是Python语法下对某个数据类型的字符串形式的结果。主要用于调试，并不用于和其他平台之间进行的数据交换。由于是主要用于调试，所以可以接受几乎一切的Python数据类型。

例如：

>>> testDict = {"A":"Apple", 'B': [1, 2, 3], "C": datetime.now()}

>>> str(testDict)

# 返回为"{'A': 'Apple', 'B': [1, 2, 3], 'C': datetime.datetime(2021, 4, 22, 11, 54, 20, 887372)}"， 可以为字符串，列表， datetime类型进行直接的Python语法下的字符串显示。**（str()函数会为整体输出结果以引号包括表示字符串，当使用print或者文件句柄.write写入文件中时，此外围引号不会存在，但内部内容相同）**

相比之下，Python的json模块是用于平台之间的数据交换，严格遵守JSON格式。json.dumps()方法返回的是满足JSON语法的字符串，使用该字符串进而和其他平台进行数据交互。而不仅仅是满足Python语法下的字符串。

并且JSON语法对数据的表示和Python语法对数据表示不尽相同。

例如：

Python语法下可以使用单引号，双引号表示字符串常量，而JSON语法只允许使用双引号。

Python语法使用None表示空对象，而JSON使用null表示空对象。

Python语法使用True表示真，False表示假，而JSON使用true表示真，false表示假。

Python语法字典可以使用多种数据类型作为字典key, 而JSON只能使用字符串作为字典key.

由于上述不同，str（某数据类型）返回的结果很有可能和json.dumps(某数据类型)返回的结束不同。str（某数据类型）返回的结果也很有可能不是有效的JSON字符串表示。所以当要与其他平台进行数据交互， 进行序列化，反序列化的时候。必须使用json.dumps()生成合法的JSON字符串，而不是以str()作为结果进行交互。

场景1：

>>> testDict = {"A":"Apple", 'B': [1, 2, 3], "C": datetime.now()}

>>> str(testDict)

"{'A': 'Apple', 'B': [1, 2, 3], 'C': datetime.datetime(2021, 4, 22, 11, 54, 20, 887372)}" # str()返回的是Python语法下的字符串显示。所以Python中的datetime类型可以直接以Python字符串形式显示。当写入文件或者print打印时候，结果为{'A': 'Apple', 'B': [1, 2, 3], 'C': datetime.datetime(2021, 4, 22, 11, 54, 20, 887372)}。当时这不是合法的JSON字符串，所以如果接下来使用json.load读取将会报错。

>>> json.dumps(testDict) # 报错，TypeError: Object of type 'datetime' is not JSON serializable。 对于JSON语法，JSON中可以接受字符串，列表，字典等数据类型。但是Python中的datetime类型不符合规定，所以不能直接转化为JSON字符串表示。需要先将Python datetime类型变为字符串后才可以。

场景2：

>>> testString = "{'A':'1'}"

>>> json.loads(testString) # 报错，Expecting property name enclosed in double quotes: line 1 column 2 (char 1). JSON语法要求字符串必须以双引号。所以testString并不是合法JSON形式字符串.

场景3：

文件内容是

{

"build": "nsp-configurator-21.6.0-rel.2446.tar.gz",

"log": "logs/iteration\_41.log",

"passedTestcases": "[‘\u2713 REGR: NSPOS: Check Fault Management via REST’]"

}

当使用json.load以后，由于是带有双引号，passedTestcases对应的Python数据类型是字符串。

而文件内容是

{

"build": "nsp-configurator-21.6.0-rel.2446.tar.gz",

"log": "logs/iteration\_41.log",

"passedTestcases": [“\u2713 REGR: NSPOS: Check Fault Management via REST”]

}

当使用json.load以后，由于是不带有双引号，而是直接使用[]，passedTestcases对应的Python数据类型是列表。