普通方法

NodeJS

- global
 - Buffer 类
 - buf.length: buffer的bytes大小
 - buf[index]: 获取或者设置在指定index索引位置的8位字节内容
 - buf.write(string, [offset], [length], [encoding]): 根据 参数 offset 偏移量和指定的encoding編码方式,将 参数 string 数据写入buffer
 - buf.toString([encoding], [start], [end]): 根据 encoding参数(默认是'utf8')返回一个解码的 string 类型
 - buf.toJSON():返回一个JSON表示的Buffer实例。 JSON.stringify将会默认调用来字符串序列化这个 Buffer实例

write

write方法写入Buffer. alloc(str)的长度是固定的;

Buffer. from(str);长度不固定

buf.write(要写入的字符串,从第几位开始写入,写入字符串长度,写入字符串的编码)

NodeJS

- Buffer 类

global

- buf.slice([start], [end]):返回一个新的buffer,这个buffer将会和老的buffer引用相同的内存地址,注意:修改这个新的buffer实例slice切片,也会改变原来的buffer
- buf.copy(targetBuffer, [targetStart], [sourceStart], [sourceEnd]): 进行buffer的拷贝

slice会改变原Buffer copy不

deals in:



类方法

NodeJS

- global
 - Buffer 类
 - Buffer.isEncoding(encoding): 如果给定的编码 encoding 是有效的,返回 true,否则返回 false。
 - Buffer.isBuffer(obj):测试这个 obj 是否是一个 Buffer.
 - Buffer.byteLength(string, [encoding]): 将会返回这个字符申真实byte长度。 encoding 编码默认是: 'utf8'
 - Buffer.concat(list, [totalLength]): 返回一个保存着 将传入buffer数组中所有buffer对象拼接在一起的 buffer对象

deals in

console.log(Buffer.isEncoding('utf-8')); console.log(Buffer.isEncoding('gbk')); 判断当前buffer是否支持这个编码 concat()连接在一起