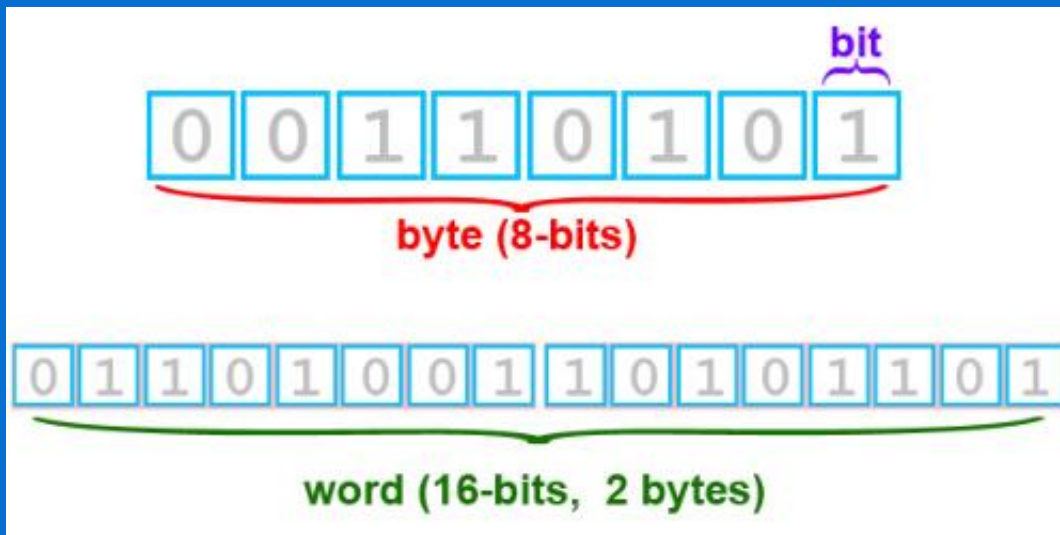


# 什么是Buffer

- 缓冲区Buffer是暂时存放输入输出数据的一段 **内存**。
- JS语言自身只有字符串数据类型，没有 **二进制** 数据类型，而在处理TCP和文件流的时候，必须要处理二进制数据。
- NodeJS提供了一个Buffer对象来提供对二进制数据的操作
- 是一个表示 **固定** 内存分配的全局对象，也就是说要放到缓存区中的字节数需要 **提前确定**
- Buffer好比由一个 **八位字节** 元素组成的数组，可以有效的在javascript中表示二进制数据

# 什么是字节

- 字节(Byte)是计算机存储时的一种 **计量** 单位，一个字节等于 **8位** 二进制数
- 一个位就代表一个0或1，每8个位 ( bit ) 组成一个字节 ( Byte )
- 字节是通过网络传输信息的单位
- 一个字节最大值十进制数是255



# ASCII 码

ASCII 码使用指定的7位或8位二进制数组合来表示128或256种可能的字

ASCII表																										
( American Standard Code for Information Interchange 美国标准信息交换代码 )																										
高四位    低四位		ASCII控制字符											ASCII打印字符													
		0000						0001					0010		0011		0100		0101		0100		0111			
		0						1					2		3		4		5		6		7			
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0 空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p		
0001	1	1	☺	^A	SOH	标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q		
0010	2	2	☹	^B	STX	正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r		
0011	3	3	♥	^C	ETX	正文结束	19	!!	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s		
0100	4	4	♦	^D	EOT	传输结束	20	¶	^T	DC4		设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t		
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u		
0110	6	6	♠	^F	ACK	肯定应答	22	—	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v		
0111	7	7	•	^G	BEL	\a 响铃	23	↕	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w		
1000	8	8	▣	^H	BS	\b 退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x		
1001	9	9	○	^I	HT	\t 横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y		
1010	A	10	◼	^J	LF	\n 换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z		
1011	B	11	♂	^K	VT	\v 纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{		
1100	C	12	♀	^L	FF	\f 换页	28	└	^_	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124			
1101	D	13	♪	^M	CR	\r 回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}		
1110	E	14	🎵	^N	SO	\s 移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~		
1111	F	15	🎵	^O	SI	\s 移入	31	▼	^.	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	␣	^Backspace 代码: DEL	
注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。 制作: MHL QQ:1208980380 2013/08/08																										

符

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。

制作：MHL QQ:1208980380 2013/08/08

峰培训  
rest training

## 定义Buffer

- `new Buffer(size)`
- 数组创建 `new Buffer(array);`
- 字符串创建 `new Buffer(str,[encoding]);`

## 字符串和Buffer转换

- Buffer转字符串 `Buffer.toString([encoding],[start],[end])`
- 字符串转Buffer `new Buffer(str,[encoding]);`

## Buffer常用方法

- 合并Buffer `Buffer.concat([buf1,buf2],length)`
- 复制Buffer  
`Buffer.copy(targetBuffer,targetstart,sourcstart,sourcend)`
- 判断是否是 `Buffer.isBuffer`
- 获取字节长度 `Buffer.byteLength`