

```

/** 在字符串末尾追加字符串 */
String.prototype.append = function (str) {
    return this.concat(str);
}
/** 删除指定索引位置的字符，索引无效将不删除任何字符 */
String.prototype.deleteCharAt = function (index) {
    if (index < 0 || index >= this.length) {
        return this.valueOf();
    }
    else if (index == 0) {
        return this.substring(1, this.length);
    }
    else if (index == this.length - 1) {
        return this.substring(0, this.length - 1);
    }
    else {
        return this.substring(0, index) + this.substring(index + 1);
    }
}
/** 删除指定索引区间的字符串 */
String.prototype.deleteString = function (start, end) {
    if (start == end) {
        return this.deleteCharAt(start);
    }
    else {
        if (start > end) {
            var temp = start;
            start = end;
            end = temp;
        }
        if (start < 0) {
            start = 0;
        }
        if (end > this.length - 1) {
            end = this.length - 1;
        }
        return this.substring(0, start) + this.substring(end + 1, this.length);
    }
}
/** 检查字符串是否以 subStr 结尾 */
String.prototype.endsWith = function (subStr) {
    if (subStr.length > this.length) {
        return false;
    }
}

```

```

        else {
            return (this.lastIndexOf(subStr) === (this.length - subStr.length)) ? true : false;
        }
    }
}
/** 比较两个字符串是否相等，也可以直接用 === 进行比较 */
String.prototype.equal = function (str) {
    if (this.length !== str.length) {
        return false;
    }
    else {
        for (var i = 0; i < this.length; i++) {
            if (this.charAt(i) !== str.charAt(i)) {
                return false;
            }
        }
        return true;
    }
}
/** 比较两个字符串是否相等，不区分大小写 */
String.prototype.equalIgnoreCase = function (str) {
    var temp1 = this.toLowerCase();
    var temp2 = str.toLowerCase();
    return temp1.equal(temp2);
}
/** 将指定的字符串插入到指定的位置后面，索引无效将直接追加到字符串的末尾 */
String.prototype.insert = function (offset, subStr) {
    if (offset < 0 || offset >= this.length - 1) {
        return this.append(subStr);
    }
    return this.substring(0, offset + 1) + subStr + this.substring(offset + 1);
}
/** 判断字符串是否数字串 */
String.prototype.isAllNumber = function () {
    for (var i = 0; i < this.length; i++) {
        if (this.charAt(i) < '0' || this.charAt(i) > '9') {
            return false;
        }
    }
    return true;
}
/** 将字符串反序排列 */
String.prototype.reserve = function () {
    var temp = "";
    for (var i = this.length - 1; i >= 0; i--) {

```

```

        temp = temp.concat(this.charAt(i));
    }
    return temp;
}
/** 将指定的位置的字符设置为另外指定的字符或字符串.索引无效将直接返回不做任何处理 **/
String.prototype.setCharAt = function (index, subStr) {
    if (index < 0 || index > this.length - 1) {
        return this.valueOf();
    }
    return this.substring(0, index) + subStr + this.substring(index+1);
}
/** 检查字符串是否以 subStr 开头 **/
String.prototype.startsWith = function (subStr) {
    if (subStr.length > this.length) {
        return false;
    }
    return (this.indexOf(subStr) == 0) ? true : false;
}
/** 计算长度，每个汉字占两个长度，英文字符每个占一个长度 **/
String.prototype.charLength = function () {
    var temp = 0;
    for (var i = 0; i < this.length; i++) {
        if (this.charCodeAt(i) > 255) {
            temp += 2;
        }
        else {
            temp += 1;
        }
    }
    return temp;
}
String.prototype.charLengthReg = function () {
    return this.replace(/[\^x00-\xff]/g, "***").length;
}
/** 去掉首尾空格 **/
String.prototype.trim = function () {
    return this.replace(/(^s*)(\s*$)/g, "");
}
/** 测试是否是数字 **/
String.prototype.isNumeric = function () {
    var tmpFloat = parseFloat(this);
    if (isNaN(tmpFloat))
        return false;

```

```

        var tmpLen = this.length - tmpFloat.toString().length;
        return tmpFloat + "0".Repeat(tmpLen) == this;
    }
    /** 测试是否是整数 */
    String.prototype.isInt = function () {
        if (this == "NaN")
            return false;
        return this == parseInt(this).toString();
    }
    /** 获取 N 个相同的字符串 */
    String.prototype.Repeat = function (num) {
        var tmpArr = [];
        for (var i = 0; i < num; i++) tmpArr.push(this);
        return tmpArr.join("");
    }
    /** 合并多个空白为一个空白 */
    String.prototype.resetBlank = function () {
        return this.replace(/s+/g, " ");
    }
    /** 除去左边空白 */
    String.prototype.LTrim = function () {
        return this.replace(/^s+/g, "");
    }
    /** 除去右边空白 */
    String.prototype.RTrim = function () {
        return this.replace(/s+$/g, "");
    }
    /** 除去两边空白 */
    String.prototype.trim = function () {
        return this.replace(/(^s+)|(s+$/g, "");
    }
    /** 保留数字 */
    String.prototype.getNum = function () {
        return this.replace(/[^\d]/g, "");
    }
    /** 保留字母 */
    String.prototype.getEn = function () {
        return this.replace(/[^A-Za-z]/g, "");
    }
    /** 保留中文 */
    String.prototype.getCn = function () {
        return this.replace(/[\u4e00-u9fa5uf900-ufa2d]/g, "");
    }
    /** 得到字节长度 */

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/896053200142010212>