

02_file基本操作

读文件

```
var fs = require('fs');

fs.readFile('./a.txt',function(err,data){
  if(!err){
    console.log(data.toString());
  }
})
```

buffer数据

```
var fs = require('fs');
//var Buffer = require('buffer');

//fd:文件描述符。指向一个文件。
//file的二进制描述:abcde
//buf:00000
fs.open('e.txt', 'a+', (err,fd)=>{
  var buf = new Buffer(5);
  fs.read(fd, buf, 0, 3, 1, (err)=>{
    if (err)
      throw err;
    console.log(buf.toString());
  });
});
```

写文件

```
fs.writeFile('./out.txt', 'Hello Node.js,我来自中国', (err) => {
  if (err)
    throw err;
  console.log('It\'s saved!');
});
```

重命名

```
var fs = require('fs');

fs.rename('./out.txt', '../e.txt', (err)=>{
  if (err)
    throw err;
  console.log('重命名好了!');
})
```

删除文件

```
var fs = require('fs');

fs.unlink('./e.txt', (err) => {
  if (err) throw err;
  console.log('successfully deleted ');
});
```

修改权限

```
var fs = require('fs');

fs.chmod('./e.txt', 777, (err) => {
  if (err) throw err;
  console.log('successfully chmod ');
})
```

访问文件权限

```
var fs = require('fs');

//测试用户对指定文件的权限。
fs.access('e.txt', fs.R_OK | fs.W_OK|fs.X_OK, (err) => {
  console.log(err ? 'no access!' : 'can read/write!');
});
```

拼接数据

```
var fs = require('fs');

fs.appendFile('e.txt', 'data to append', (err) => {
  if (err) throw err;
  console.log('The "data to append" was appended to file!');
});
```

查看文件信息

```
var fs = require('fs');

fs.watchFile('e.txt', (curr, prev) => {
  console.log(`the current 修改时间(mtime) is: $
{curr.mtime}`);
  console.log(`the previous mtime was: ${prev.mtime}`);
});
```

查看目录信息

```
var fs = require('fs');

fs.watch('www', (event, filename) => {
  console.log(`事件是 is: ${event}`);
  if (filename) {
    console.log(`filename provided: ${filename}`);
  } else {
    console.log('filename not provided');
  }
});
```

读取目录下的文件

```
fs.readdir(path, function (err, files) {
  files.forEach(function (file, i) {
    console.log(file, i)
  })
})
```

1. 把path路径下的所有文件读取出来, 并拼接,回调到fn函数中

```
function readContentFromPath(path,fn){
  var allStr = '';

  fs.readdir(path,function(err,files){
    //console.log(files);
    files.forEach(function(file,i){
      fs.readFile(path+'/'+file,function(err,content){
        if(err)
          throw err;
        allStr += content;
        // console.log(content.toString());
        if(i == files.length-1) {
          fn(allStr);
        }
      })
    })
  })
}
```

2. 替换空格和回车等字符,进行文件的压缩

```
//process.argv 是一个数组,里面包含,node的路径,当前js的路径,和其他传进来的参数。
var path = process.argv[2];//通过process,拿到压缩路径

readContentFromPath('./src',function(content){
  content = content.replace(/\s/g,'');
  content = content.replace(/\n/*[\s\S]*?\\n/g,'');
  fs.writeFile('./dist.js',content,function(err){
    if(err)
      throw err;
    console.log('压缩完成。。');
  })
})

fs.watch(path,function(){
  console.log('文件有变化...');
  compress('./dist.js');
})
```

3. 一个简单的css,js压缩工具

<https://www.npmjs.com/package/uglify>

判断某一个文件是否存在,并且是否是img

```
var path = "xxx路径"
//查看path路径是否存在
fs.access(path, fs.R_OK, (err) => {
  if (err){
    fn('文件不存在');
  }else{
    //console.log('读取文件: '+path)
    //读取文件
    fs.readFile(path, function (err, data) {
      if (err)
        throw err;
      console.log('读取文件: '+data)
      //给文件添加mime类型。
      if( /(gif|jpg)$/.test(path)){
        var type = 'image/jpeg'
      }
      type?fn(data,type):fn(data.toString(),type)
    })
  }
});
```