

[| 카카오톡 봇 강좌 | >](#)

[중급] (라이노 엔진) 자바스크립트 기초 강의 (7 - 1) - 파일 입출력 및 기타 등등 (java.io 패키지)



마른얼음 BOT 1:1 채팅

2020.01.14. 16:07 조회 439

💬 댓글 26 URL 복사 ⋮

(주의) 이 강의의 길이는 제 다른 강의보다 역대급으로 길 수 있습니다.

오랜만에 써보는 강의군요! 그만큼 저도 매우 중요한 강의 주제를 들고 왔습니다. 이번에는 java.io.File 클래스와 InputStream, OutputStream 등등을 활용한 파일 입출력을 해 보고, 또 관련된 여러 파일들을 다루는 방법을 알려드리겠습니다. 참고로 java.io.File 클래스는 자바 기준으로 만들어져있기 때문에, 약간의 자바 지식이 없으면 이해하는데 약간의 어려움이 있을 것입니다.

일단 java.io.File 클래스는 생성자를 통해 파일의 경로를 설정해 줄 겁니다. (뭔 말인지 모르겠다면 이 문장은 스킵하셔도 좋습니다)

```
new java.io.File(path);
```

path(경로)는 문자열이어야 합니다! 저는 예시로 sdcard 안에 있는 FunnyJS 폴더를 경로로 삼아보죠.

```
var myFile = new java.io.File("/sdcard/FunnyJS");
```

저희의 목표는, FunnyJS 폴더 안에 FunnyJS.txt 파일을 생성한 뒤, response 함수를 이용해 '/파일입력 (값)' 메시지로 입력받은 값을 그 파일 안에 입력한 뒤, '/파일보기' 메시지를 받았을 때 FunnyJS.txt 파일 내용을 읽어올 겁니다. 더불어 '/파일삭제' 메시지를 받았을 때는 파일을 (숙청)삭제해버릴 겁니다.

일단 위에 해당하는 틀을 잡아줄까요?

```
function response(room, msg, sender, isGroupChat, replier) {  
  if (msg.indexOf("/파일입력 ") == 0) {  
    //파일 생성 및 내용 쓰기  
  }  
  if (msg == "/파일보기") {  
    //파일 내용 답장하기  
  }  
  if (msg == "/파일삭제") {  
    //파일 삭제하고 그 여부 답장하기  
  }  
}
```

자, 이제 틀은 잡혔습니다! 이제 파일 관련 애들을 만들어봅시다. 우선 파일을 생성해볼까요? 파일 생성 관련 메소드들을 가져와봤습니다.

```
new java.io.File(path).exists();  
new java.io.File(path).mkdir();  
new java.io.File(path).mkdirs();  
new java.io.File(path).createNewFile();
```

우리가 파일 생성에서 주로 쓸 것들은 요 네가지 입니다.

일단 첫 번째 메소드 `exists`는 메소드 이름에서 보이듯이 파일이 존재하는지에 대한 논리값(`true`, `false`)를 반환합니다.

그리고 두 번째와 세 번째 메소드들은 폴더를 생성하는 메소드로, 언뜻 보면 거기서 거기인 듯 싶지만 전혀 다릅니다!
두 번째 메소드 `mkdir`는, 생성하려는 폴더의 상위 폴더가 없으면 `false`를 반환하고, 그렇지 않으면 폴더를 생성한 후 `true`를 반환합니다.

세 번째 메소드 `mkdirs`는, 생성하려는 폴더의 상위 폴더가 있는 없는 필요한 폴더들을 무조건 다 생성해줍니다.

그리고 네 번째 메소드 `createNewFile`은 역시 이름에서 보이듯이 파일을 생성해줍니다.

그럼 이제 `FunnyJS` 폴더 안에 `FunnyJS.txt` 파일을 만들어서 파일 입력까지 해봅시다!

먼저 `FunnyJS` 폴더와 `FunnyJS.txt` 파일을 만들어줍니다.

```
var myFile = new java.io.File("/sdcard/FunnyJS/FunnyJS.txt");
if (myFile.exists() == false) {
    myFile.getParentFile().mkdirs();
    myFile.createNewFile();
}
```

여기서 뭔가 처음보는 것이 등장할 것입니다.

```
myFile.getParentFile().mkdirs();
```

어? 이게 뭐지?

일단 맨 뒤에 있는 `mkdirs`로 폴더를 모두 생성해준다는건 알겠는데, 중간에 `getParentFile`은 뭘까요? 애는 바로 상위 경로를 담고 있는 `File` 객체를 반환합니다. 왜 이런식으로 파일을 생성하는 것일까요?

아까 위를 자세히 보면 알 수 있습니다. `mkdirs` 메소드는 폴더를 만드는 메소드이기 때문에 파일은 만들지 못합니다. 그래서 파일 이름인 `FunnyJS.txt`조차도 폴더로 만들어버릴 수 있습니다! (ㅎㄷㄷ) (내가 이거때문에 카톡봇 시물 만들때 개고생을 했는데 읊음)

그래서 일부러 상위 경로를 가져와서 폴더를 먼저 만들어준 뒤, 파일을 생성해주는 겁니다.

여튼 파일이 존재하지 않으면 파일을 생성해준다는 그런 내용이었네요.

이 다음, 이제 파일에 내용을 넣어 줄 겁니다. 파일에 내용을 입력할 때는 `OutputStream`을 상속받은 `FileOutputStream`을 이용하는데요, 이 녀석은 파일에 연결해서 데이터를 주는 형식으로 작동합니다. 생성자를 통해 파일 경로를 받고 `write` 메소드를 이용해 내용을 넣습니다. 제가 한번 사용 예시를 보여드리겠습니다.

```
var str = "파일에 입력할 내용이 들어갑니다.";
var fos = new java.io.FileOutputStream(myFile);
fos.write(new java.lang.String(str).getBytes());
fos.close();
```

음.. 살짝 복잡해보이긴 하지만 간단합니다.

일단 `FileOutputStream`은 매개변수를 `java.io.File` 또는 `String`으로 받습니다. (`FileDescriptor`를 받기도 하는데 그건 생략하겠습니다.) 즉 위 예제처럼 `File` 객체를 넣어줘도 되고 그냥 문자열로 파일의 경로를 넣어도 된다는 겁니다.

그리고 `write` 하는 부분에서 `java.lang.String(str).getBytes()` 이라는것이 등장하는데요, 이것은 문자열을 바이트 배열로 변환하기 위한 작업입니다. `write`는 인자를 바이트로 받기 때문이죠.

그리고 마지막으로 `FileOutputStream` 사용이 끝났으니 `close`를 통해 파일과의 연결을 끊습니다.

자 이제 지금까지 짜놓았던 소스들을 합쳐봅시다!

```
var myFile = new java.io.File("/sdcard/FunnyJS/FunnyJS.txt");
if (myFile.exists() == false) {
    myFile.getParentFile().mkdirs();
```

```

myFile.createNewFile();
}
var str = "파일에 입력할 내용이 들어갑니다.";
var fos = new java.io.FileOutputStream(myFile);
fos.write(new java.lang.String(str).getBytes());
fos.close();

```

소스가 조금 길군요. 이것을 response 함수에 통째로 넣어줘도 되지만, 소스 가독성을 고려하여 저는 함수 형태로 만들어주겠습니다. (함수를 모르시는 분들은 제 강의 2강을 참고하시면 좋을 것 같습니다)

```

function saveFile(path, value) {
var myFile = new java.io.File(path);
if (myFile.exists() == false) {
myFile.getParentFile().mkdirs();
myFile.createNewFile();
}
var fos = new java.io.FileOutputStream(myFile);
fos.write(new java.lang.String(value).getBytes());
fos.close();
}

```

파일의 경로와 입력할 내용을 매개변수로 넘겨줬습니다. 이러면 아무래도 사용 범위가 넓어져서 다른 기능을 만들 때 활용할 수 있겠네요.

그럼 이제 다음 할 일을 찾아봅시다! 다음 할 일은 파일 내용 담장하기네요. 이번에는 파일 내용을 읽는 방법을 알아보시다.

파일을 읽을 때에는 InputStream 클래스를 상속받은 FileInputStream을 사용하는데, 이 녀석만 사용하는건 비효율적입니다. 그래서 우리는 BufferedReader를 사용할겁니다. 이 녀석은 버퍼를 이용하여 데이터를 전달하는데, 우리가 짐을 나른다고 생각해 봅시다. 만약 버퍼를 사용하지 않고 짐을 나른다면 우리는 짐을 한 개씩 한 개씩 나를 것입니다. 그런데 버퍼를 사용하면, 짐을 모두 버퍼라는 상자에 담아서 짐을 한 번 또는 몇여 번에 걸쳐서 짐을 다 나를 수 있습니다. 그래서 이 BufferedReader는 보다 더 효율적입니다.

일단 한 번 소스를 만들어 봅시다.

```

var myFile = new java.io.File("/sdcard/FunnyJS/FunnyJS.txt");
var fis = new java.io.FileInputStream(myFile);
var isr = new java.io.InputStreamReader(fis);
var br = new java.io.BufferedReader(isr);
var result = br.readLine();
var test = "";
while ((test = br.readLine()) != null) {
result += "\n" + test;
}
fis.close();
isr.close();
br.close();

```

소스가 좀 길죠? 원래 버퍼를 사용하는 소스를 짜면 이렇게 번거롭습니다. 하지만 그만큼 속도가 빨라집니다!

천천히 소스를 분석해봅시다. 일단 FileInputStream도 아까 전 FileOutputStream과 똑같이 매개변수를 java.io.File 또는 문자열로 받습니다.

그리고 BufferedReader는 매개변수를 Reader로 받습니다. 그래서 우리는 FileInputStream에 맞는 InputStreamReader를 사용했습니다.

그리고 BufferedReader는 띄어쓰기 단위 또는 개행문자(엔터) 단위로 끊어서 데이터를 버퍼에 저장하는

그리고, `BufferedReader`은 리어쓰기 단위 또는 개행문자(엔터) 단위로 끊어서 데이터를 보내는데, 여기서도 `br.readLine()` 을 이용하여 개행문자(엔터) 단위로 끊어서 데이터를 받아왔습니다. 엔터 단위로 끊어서 데이터를 계속 읽어와서 읽어온 데이터가 null 일 때 까지 `result` 변수에 넣어주세요.

그리고 사용이 끝난 `InputStream`, `Reader`들은 다 `close`를 통해 연결을 끊습니다.

이제 애도 함수로 만들어볼까요?

```
function readFile(path) {
    var myFile = new java.io.File(path);
    var fis = new java.io.FileInputStream(myFile);
    var isr = new java.io.InputStreamReader(fis);
    var br = new java.io.BufferedReader(isr);
    var result = br.readLine();
    var test = "";
    while ((test = br.readLine()) != null) {
        result += "\n" + test;
    }
    fis.close();
    isr.close();
    br.close();
    return result;
}
```

역시 이것도 경로를 매개변수로 넘겨줬습니다.

자 이제 마지막 할 일은? 파일 삭제하고 그 여부 답장하기 입니다.

파일을 삭제하는 메소드는 단 한개입니다!

```
new java.io.File(path).delete();
```

파일을 삭제하고 그 여부를 보내줍니다.

이제 애를 다이렉트로 함수로 만들어줍시다!

```
function removeFile(path) {
    return new java.io.File(path).delete();
}
```

이제 모든 준비가 다 되었습니다! 이제 소스를 완성시켜봅시다.

```
function response(room, msg, sender, isGroupChat, replier) {
    if (msg.indexOf("/파일입력 ") == 0) {
        var stringOfFile = readFile("/sdcard/FunnyJS/FunnyJS.txt") + msg.substr(6);
        saveFile("/sdcard/FunnyJS/FunnyJS.txt", stringOfFile);
        replier.reply("파일 입력 완료.");
    }
    if (msg == "/파일보기") {
        replier.reply("파일을 읽어왔습니다: \n\n" + readFile("/sdcard/FunnyJS/FunnyJS.txt"));
    }
    if (msg == "/파일삭제") {
        var isDeleted = removeFile("/sdcard/FunnyJS/FunnyJS.txt");
        if (isDeleted == true) {
            replier.reply("파일이 삭제되었습니다.");
        } else {
            replier.reply("파일이 삭제되지 않았습니다.");
        }
    }
}
```

```
}  
}
```

소스가 완성된 것 같군요! 그럼 지금까지 만든 소스들을 정렬해봅시다. 그리고 저 파일이 있는 경로는 계속 반복되니 상수로 만들어 버리겠습니다.

```
const PATH = "/sdcard/FunnyJS/FunnyJS.txt";
```

```
function saveFile(path, value) {  
  var myFile = new java.io.File(path);  
  if (myFile.exists() == false) {  
    myFile.getParentFile().mkdirs();  
    myFile.createNewFile();  
  }  
  var fos = new java.io.FileOutputStream(myFile);  
  fos.write(new java.lang.String(value).getBytes());  
  fos.close();  
}
```

```
function readFile(path) {  
  var myFile = new java.io.File(path);  
  var fis = new java.io.FileInputStream(myFile);  
  var isr = new java.io.InputStreamReader(fis);  
  var br = new java.io.BufferedReader(isr);  
  var result = br.readLine();  
  var test = "";  
  while ((test = br.readLine()) != null) {  
    result += "\n" + test;  
  }  
  fis.close();  
  isr.close();  
  br.close();  
  return result;  
}
```

```
function removeFile(path) {  
  return new java.io.File(path).delete();  
}
```

```
function response(room, msg, sender, isGroupChat, replier) {  
  if (msg.indexOf("/파일입력 ") == 0) {  
    var stringOfFile = readFile(PATH) + msg.substr(6);  
    saveFile(PATH, stringOfFile);  
    replier.reply("파일 입력 완료.");  
  }  
  if (msg == "/파일보기") {  
    replier.reply("파일을 읽어왔습니다: \n\n"+readFile(PATH));  
  }  
  if (msg == "/파일삭제") {  
    var isDeleted = removeFile(PATH);  
    if (isDeleted == true) {  
      replier.reply("파일이 삭제되었습니다.");  
    } else {  
      replier.reply("파일이 삭제되지 않았습니다.");  
    }  
  }  
}
```

```
}  
}  
}
```

소스 길이 역대급..

우리가 이렇게 파일을 다루는 함수들을 3개 만들어봤는데, 사실 달토봇/메신저봇에서는 이 함수들이 이미 존재합니다!
(개고생)

그래서 다음 강의에는 그것들에 대해 다뤄보도록 하겠습니다.

그럼 강의를 마치겠습니다.

정말정말 오랜만입니다!

이번에는 내용이 조금 길었네요. 자바 클래스 개념이 약간 들어가 있어서 이해하기 약간 어려우실 분들도 있을 것 같아
후회되네요.. 그래도 저는 최선을 다해서 설명했으니 이해 부탁드립니다 ㅠ

그럼 이만



마른얼음 BOT님의 게시물 더보기 >

❤️ 좋아요 6 💬 댓글 26

🔗 공유 | 신고

댓글 등록순 최신순 🔒

댓글알림 ☐



별명을 입력해주세요

라는 거시야요 ..!



2020.01.14. 16:15 답글 쓰기



마른얼음 작성자

띠용

2020.01.14. 16:18 답글 쓰기



프로그

그렇지 않아도 파일스트림에서 java.io로 갈아타려했는데 마땅한 강좌가 없어 지금까지 못하고 있었네요. 강좌 감사합니다!

2020.01.14. 16:21 답글 쓰기



마른얼음 작성자

오... 도움이 되신다니 진짜 감사드립니다



2020.01.14. 16:22 답글 쓰기



SP

어버버버 좀더 공부하고 올게요

2020.01.14. 16:52 답글 쓰기



마른얼음 작성자

허허.. 좋습니다

2020.01.14. 16:58 답글 쓰기



SP

마른얼음 그래도 하트는 누릅니다 허허..

2020.01.14. 17:07 답글쓰기



scap



2020.01.14. 16:57 답글쓰기



마른얼음 작성자

와!



2020.01.14. 16:58 답글쓰기



현 흑

아직 머리에만 두고 안해봤는데 소켓통신도 되려나요?

2020.01.14. 18:50 답글쓰기



현 흑

소켓통신되면 가가라이브 소켓연결시켜서 봇으로 가가라이브 연동 가능한데

2020.01.14. 18:51 답글쓰기



마른얼음 작성자

OutputStream과 InputStream을 적절히 활용한다면 아마 될 겁니다. java.io 패키지에 있는 애들은 거의 다 입출력 관련 클래스들이라서..

2020.01.14. 18:51 답글쓰기



현 흑

마른얼음 네 이론상은 가능해보이는데 해봐야 알겠는데 시간이 없네요.

2020.01.14. 18:55 답글쓰기



scap

java.net.Socket

2020.01.14. 19:24 답글쓰기



사로로

원래 파일 입출력이 예외처리 안하면 음음

2020.01.14. 20:20 답글쓰기



마른얼음 작성자

아 맞다 예외처리 ㄸㄸ..

2020.01.14. 20:20 답글쓰기



아미타

좋은 글 감사합니다.

2020.01.15. 14:38 답글쓰기



마른얼음 작성자

저도 뿌듯하네용



2020.01.15. 19:17 답글쓰기



NotreDame

와... 세상에나 마상에나

2020.01.15. 19:17 답글쓰기



마른얼음 작성자

?????????

2020.01.15. 19:18 답글쓰기



NotreDame

마른얼음 흑..님의 경지에 오르려면 얼마나 더 해야할까요..ㅠㅠ 한 1년이면 되려나... 존경합니다 갓 드라이아이스

2020.01.15. 19:18 답글쓰기



AlphaDo

나 문서 왜 읽었지

2020.01.16. 13:32 답글쓰기



마른얼음 작성자

ㄷ ㄷ

2020.01.16. 13:32 답글쓰기



김각제

항상 고마워요!! 매번 신세지네요 감사합니다!!

2020.02.13. 20:26 답글쓰기

삭제된 댓글입니다.



scap

mkdirs는 폴더 생성입니다

2020.03.10. 07:28 답글쓰기



ailla

문서 이름뒤에 .txt 꼭 붙여야 하나요?

2020.03.10. 19:15 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'| 카카오톡 봇 강좌 |' 게시판 글

전체 [중급] 말머리 글

이 게시판 새글 구독하기 ☐

[초급] with 구문이란? [1]	마른얼음	2020.01.19.
[초급] (라이노 엔진) 자바스크립트 기초 강의 (7 - 2) - 파일 입출력 및 기타 등등 (FileStream 라이브러리) [27]	마른얼음	2020.01.14.
[중급] (라이노 엔진) 자바스크립트 기초 강의 (7 - 1) - 파일 입출력 및 기타 등등 (java.io 패키지) [26]	마른얼음 BOT	2020.01.14.
[팁] Wu202E 없애기 [4]	알랑뽕까	2020.01.09.
온라인 카카오톡 트래커 #2-2 🤖 [7]	delta	2020.01.09.

1 2 3

전체보기

이 카페 인기글

카톡방 나가기

hajuhee01
 ❤️ 0 💬 15



혹시 여기서 변수를 저장하는 부분이 어딘지
아시는분 계시나요??
공식 카카오톡 오픈채팅
Milk2

Lunar  4
 0  9

간편 자동응답과 한글코딩을 메인으로 제공
하는 카톡봇앱 개발 예정

성빈
♡1 💬16



태양, 달 정보 구현 완료



네이버에서 지역별 날씨 정보 크롤링하기 /

1 2 3 4 5

안녕하세요

tomohong
♡ 1 💬 3

[치***, 한**] 님 강제 탈퇴

AlphaDo
♥ 1 💬 2

안녕하세요

0 5