

| 카카오톡 봇 강좌 | >

[중급] (라이노 엔진) 자바스크립트 기초 강의 (5 - 2) - 파싱 예제 첫 번째, 날씨

마른얼음 BOT 1:1 채팅

2019.09.01. 09:41 조회 814

댓글 15 URL 복사

(5 - 2 ~ 5 - X 강좌는 파싱 예제이므로 자신이 파싱을 잘 한다 싶으시면 스킵하셔도 좋습니다.)

이번시간에는 한번 날씨를 뜯어와봅시다.

먼저 `Utils.getWebText`를 이용하는 방식으로 해봅시다.

먼저, **홍성** 날씨가 있는 HTML을 가져와봅시다. 왜 굳이 **홍성**이냐구요? 그냥요(?)

(편의성을 위해 변수 부분만 쓰겠습니다.)

```
var weather = Utils.getWebText("https://m.search.naver.com/search.naver?query=홍성+날씨");
```

[복사](#) [공유](#) [모두 선택](#) [번역](#)

홍성 폭염경보</div><div class="weather_bx"><div class="wt_top"><div class="tlt_wrap"><div class="wt_tlt"><i class="icon_slt"></i><h3>충청남도 홍성군 홍성읍</h3></div></div><div class="wt_opt"><ul class="_wt_opt_ul">내위치<li class="_alarm" style="display:none">알림</div><div class="wt_tab_nav_bx"><ul class="lst" role="tablist"><li class="_tab on" aria-selected="true" role="tab" data-blind-text="오늘" data-maintab="today"><a nocr onclick="goOtherTCR(this,'a=nco_x7t*4.tmb&r=1&i=88157038_00000065035F');return false;" href="#">오늘<li class="_tab" role="tab" data-blind-text="내일" data-maintab="tomorrow"><a nocr onclick="goOtherTCR(this,'a=nco_x7t*4.mmb&r=1&i=88157038_00000065035F');return

```

false;" href="#"><span class="menu">내일</span></a> </li> <li
class="_tab" role="tab" data-blind-text="모레" data-
maintab="aftertomorrow"> <a nocr onclick="goOtherTCR(this,
'a=nco_x7t*4.amb&r=1&i=88157038_00000065035F');return
false;" href="#"><span class="menu">모레</span></a> </li> <li
class="_tab" role="tab" data-blind-text="월간" data-
maintab="monthly"> <a nocr onclick="goOtherTCR(this,
'a=nco_x7t*4.mob&r=1&i=88157038_00000065035F');return
false;" href="#"><span class="menu">월간</span></a> </li> </
ul> </div> </div> <div class="wt_info"> <div id="mflick"

```

상태가 지저분하니 필요없는곳은 날려버립시다. <div class="wt_top"> 기준으로 앞 부분은 자르겠습니다.

```

var weather = Utils.getWebText("https://m.search.naver.com/search.naver?query=홍성+날씨").split("<div
class=W\"wt_topW\">")[1]

```

```

class="flick-ct eg-flick-panel" style="display:block"> <div
class="wt_status"> <div class="status_bx"> <div
class="status"> <div class="wt_img"> <span class="ico_status
ws1"><span class="u_hc">맑음</span></span> </div> <div
class="wt_text"> <strong><span class="u_hc">현재온도</
span><em class="figure degree_code">33</em></strong>
<span class="chill_temp"><span>체감온도</span><em
class="figure degree_code">35.5</em></span> </div> </div>
<div class="wt_summary"> <p class="main">맑음, 어제보다 <em
class="figure degree_code">3</em> 높아요</p> <a nocr
onclick="return goOtherCR(this,
'a=nco_x7t*4.pdst&r=1&i=88157038_00000065035F&u=' +
urlencode(this.href));"
href="?where=m&sm=mtb_etc&query=%EC%B6%A9%EC%B2%
AD%EB%82%A8%EB%8F%84%20%ED%99%8D%EC%84%B1%E
A%B5%B0%20%ED%99%8D%EC%84%B1%EC%9D%8D%20%EB
%AF%B8%EC%84%B8%EB%A8%BC%EC%A7%80"
class="link_dust"> <span>미세먼지</span><em class="figure
sign1">8<span>μg/㎥</span></em> </a> </div> <div
class="status_list"> <ul> <li><strong>습도 </strong><span
class="figure sign1">71</span></li> <li><strong>외선 </
strong><span class="figure sign1">강</span></li> <li><strong>
강풍속 </strong><span class="figure sign1">10</span></li> </ul> </div>
</div> <div class="grap_box"> <div class="title_box
_sub_nav_bx"> <h3>시간별 예보</h3> <div class="title_sub"> <a
nocr onclick="goOtherTCR(this,
'a=nco_x7t*4.wth&r=1&i=88157038_00000065035F');return
false;" href="#" class="sub_option_tab" role="option" data-
subtab="weather" aria-selected="true">날씨</a> <a nocr
onclick="goOtherTCR(this,
'a=nco_x7t*4.pre&r=1&i=88157038_00000065035F');return
false;" href="#" class="sub_option_tab" role="option" data-
subtab="rainfall">강수</a> <a nocr onclick="goOtherTCR(this,
'a=nco_x7t*4.his&r=1&i=88157038_00000065035F');return
false;" href="#" class="sub_option_tab" role="option" data-
subtab="history">기상

```


여전히 지지분하군요! 시간별 예보 기준으로 뒷 부분은 자르겠습니다.

복사 공유 모두 선택 번역

```
class="flick-ct eg-flick-panel" style="display: block" > <div
class="wt_status"> <div class="status_bx"> <div
class="status"> <span class="wt_img"> <span class="ico_status
ws1"><span class="u_hc">맑음</span></span> </div> <div
class="wt_text"> <strong><span class="u_hc">현재온도</
span><em class="figure degree_code">33</em></strong>
<span class="chill_temp"><span>체감온도</span><em
class="figure degree_code">35.5</em></span> </div> </div>
<div class="wt_summary"> <p class="main">맑음, 어제보다 <em
class="figure degree_code">3</em> 높아요</p> <a nocr
onclick="return goOtherCR(this,
'a=nco_x7t*4.pdst&r=1&i=88157038_00000065035F&u=' +
urlencode(this.href));"
href="?where=m&sm=mtb_etc&query=%EC%B6%A9%EC%B2%
AD%EB%82%A8%EB%8F%84%20%ED%99%8D%EC%84%B1%E
A%B5%B0%20%ED%99%8D%EC%84%B1%EC%9D%8D%20%EB
%AF%B8%EC%84%B8%EB%A8%BC%EC%A7%80"
class="link_dust"> <span>미세먼지</span><em class="figure
sign1">8<span>μg/ m³</span></em> </a> </div> <div
class="status_list"> <ul> <li><strong>습도 </strong><span
class="percent">71</span></li> <li><strong>자외선 </
strong><span>매우높음9</span></li> <li><strong>풍속 </
```

아까보다는 나아졌지만, 그래도 앞 부분에 쓸데없는 부분이 있네요. <div class="status_bx"> 부분을 기준으로 앞부분을 자릅시당

 또는 사이에 정보들이 끼어있네요! 편하게 할 수 있도록 또는 을 기준으로 배열로 만들어줍니다. 아 참고로, split에서도 정규식을 사용할 수 있습니다!

[봇] <div class="status"> <div class="wt_img"> 맑음</div> <div class="wt_text">,현재온도<em class="figure degree_code">32,체감온도<em class="figure degree_code">35.2 </div> </div> <div class="wt_summary"> <p class="main">맑음, 어제보다 <em class="figure degree_code">3 높아요</p> <a noc


```

onclick="return goOtherCR(this,
'a=nco_x7t*4.pdst&r=1&i=88157038_00000065035F
&u=' + urlencode(this.href));"
href="?where=m&sm=mtb_etc&query=%EC%B6%A9%
EC%B2%AD%EB%82%A8%EB%8F%84%20%ED%99%8
D%EC%84%B1%EA%B5%B0%20%ED%99%8D%EC%8
4%B1%EC%9D%8D%20%EB%AF%B8%EC%84%B8%EB
%A8%BC%EC%A7%80" class="link_dust">
<span>미세먼지</span><em class="figure
sign1">8<span>μg/m³</span></em> </a> </div> <div
class="status_list"> <ul> <li>,습도 ,<span
class="percent">72</span></li> <li>,자외선
,<span>매우높음9</span></li> <li>,풍속 ,<span
class="metersec">서 3</span></li> </ul> </div> </
div> <div class="grap_box"> <div class="title_box
_sub_nav_bx"> <h3>

```

저어기 쉼표들이 보이시나요? 저 쉼표가 배열의 쉼표를 의미합니다. (맑음, 은 빼고) 보니까 쉼표 기준 0번째는 상태(맑음), 1번째는 현재온도, 2번째는 체감온도가 나와있습니다. 우선 0번째를 가져와서 태그들 모두 없애고 trim으로 공백을 날려버립니다.

```

var weather = Utils.getWebText("https://m.search.naver.com/search.naver?query=홍성+날씨").split("<div
class=W\"wt_topW\">")[1].split("시간별 예보")[0].split("<div class=W\"status_bxW\">")[1].split("<strong>|<W/strong>/g");
var status = weather[0].replace(/<[^>]+>/g,"").trim();

```

```

[Debug] var weather = Utils.getWebText("https://
m.search.naver.com/
search.naver?query=홍성+날씨").split("<div
class=\"wt_top\">")[1].split("시간별 예보")[0].split("<div
class=\"status_bx\">")[1].split("<strong>|</strong>/
g");
weather[0].replace(/<[^>]+>/g,"").trim()
[봇] 맑음

```

좋네요! 그럼 이제 1번째것도 가져와서 똑같이 해봅시다. 1번째건 현재온도였죠? 현재온도를 읽을 때 가독성을 높이기 위해 현재온도와 온도 사이에 콜론 (:) 을 넣어주고, 끝에 단위도 붙여줍시다

```

var weather = Utils.getWebText("https://m.search.naver.com/search.naver?query=홍성+날씨").split("<div
class=W\"wt_topW\">")[1].split("시간별 예보")[0].split("<div class=W\"status_bxW\">")[1].split("<strong>|<W/strong>/g");
var nowTemp = weather[1].replace(/<[^>]+>/g,"").replace("현재온도","현재온도 : ").trim() + "°C";

```

```

[Debug] var weather = Utils.getWebText("https://
m.search.naver.com/
search.naver?query=홍성+날씨").split("<div
class=\"wt_top\">")[1].split("시간별 예보")[0].split("<div
class=\"status_bx\">")[1].split("<strong>|</strong>/
\

```

```
g);
weather[1].replace(/<[^>]+>/
g,"").replace("현재온도","현재온도 : ").trim() + "°C";
[봇] 현재온도 : 31°C
```

(저 위에서 본 사진과 지금 사진의 온도가 다른 이유는 힘들어서 미뤄가면서 강좌를 썼기 때문..크흠)

좋습니당! 이제 체감온도를 가져옵시다. 체감온도는 미세먼지와 같이 2번째에 있었죠? 미세먼지를 자르기 위해 2번째를 다시 봐봅시다.

```
[봇] <span class="chill_temp"><span>체감온도</
span><em class="figure degree_code">35.2</em></
span> </div> </div> <div class="wt_summary"> <p
class="main">맑음, 어제보다 <em class="figure
degree_code">3</em> 높아요</p> <a nocr
onclick="return goOtherCR(this,
'a=nco_x7t*4.pdst&r=1&i=88157038_00000065035F
&u=' + urlencode(this.href));"
href="?where=m&sm=mtb_etc&query=%EC%B6%A9%
EC%B2%AD%EB%82%A8%EB%8F%84%20%ED%99%8
D%EC%84%B1%EA%B5%B0%20%ED%99%8D%EC%8
4%B1%EC%9D%8D%20%EB%AF%B8%EC%84%B8%EB
%A8%BC%EC%A7%80" class="link_dust">
<span>미세먼지</span><em class="figure
sign1">8<span>µg/m³</span></em> </a> </div> <div
class="status_list"> <ul> <li>
```

우리는 어제보다 몇 도 높다는건 뽑아오지 않을겁니다. 그러니 체감온도는 에서 잘라서 앞부분을 가져 오고, 미세먼지는 </p>에서 잘라서 뒷부분을 가져옵시다. 또 체감온도의 가독성을 높이기 위해 사이에 콜론을 넣어주고 단위도 붙여줍시다. 미세먼지는 단위가 이미 붙여져있기 때문에 콜론만 넣어줍시다.

```
var weather = Utils.getWebText("https://m.search.naver.com/search.naver?query=홍성+날씨").split("<div
class=W\"wt_top\">")[1].split("시간별 예보")[0].split("<div class=W\"status_bx\">")[1].split("<strong>|<W/strong>/g);
var chillTemp = weather[2].split("</em></span>")[0].replace(/<[^>]+>/g,"").replace("체감온도","체감온도 : ").trim()
+ "°C";
var dust = weather[2].split("</p>")[1].replace(/<[^>]+>/g,"").replace("미세먼지","미세먼지 : ").trim();
```

자 이제 테스트를 해볼까요?

```
[Debug] var weather = Utils.getWebText("https://
m.search.naver.com/
search.naver?query=홍성+날씨").split("<div
class=\"wt_top\">")[1].split("시간별 예보")[0].split("<div
class=\"status_bx\">")[1].split("</strong>|</strong>/
g);
var chillTemp = weather[2].split("</em></
span>")[0].replace(/<[^>]+>/
```

```
g,"").replace("체감온도","체감온도 : ").trim() + "°C";
var dust = weather[2].split("</p>")[1].replace(/<[^>]+>/g,"").replace("미세먼지","미세먼지 : ").trim();
chillTemp + "\n" + dust
[봇] 체감온도 : 33.2°C
미세먼지 : 21µg/m³
```

잘 되니 이제 최종적으로 소스를 적당히 짜봅시다. /날씨 (지역명)을 입력하면 날씨가 나오도록 하게 해봅시다!

(실력 상승을 위해 직접 소스를 짜 봅시다! ^^)

이제 잘 작동되나 확인해볼까요?

```
[Debug] /날씨 서울
[봇] [서울 지역의 날씨]
상태 : 맑음
현재온도 : 33°C
체감온도 : 35.9°C
미세먼지 : 22µg/m³
[Debug] /날씨 수원
[봇] [수원 지역의 날씨]
상태 : 맑음
현재온도 : 34°C
체감온도 : 36.7°C
미세먼지 : 22µg/m³
[Debug] /날씨 환상의나라
```

잘 되네요! 하지만 환상의나라같은 없는 곳을 입력한다면 오류를 뿜네요. 이때 우리는 try 를 이용할겁니다. try가 뭔지 잠깐 알려드리자면,

```
try { /*소스1*/ } catch (e) { /*소스2*/ } finally { /*소스3*/ }
```

일단 try 뒤의 중괄호 사이에 있는 소스, 즉 소스1을 실행시켜줍니다. 그리고 소스1에서 만약 오류가 난다면 catch 뒤 중괄호에 있는 소스, 즉 소스2를 실행시킵니다. 아 참고로 catch 뒤에 e라는게 있죠? 이 e는 소스1에서 난 에러를 담은 변수입니다.

그리고 오류가 났든 안 났든 finally 뒤에 있는 소스3을 실행시킵니다. 참고로 이 finally는 안 써도 됩니다.

썰든 이 try catch 문을 이용하여 소스를 다시 짜봅시다.

```
function response(room, msg, sender, isGroupChat, replier) {
  if (msg.indexOf("/날씨 ") == 0) {
    try {
      //실력 상승을 위해 직접 소스를 짜 봅시다! ^^
    } catch (e) {
      replier.reply("에러가 발생했습니다. 지역명을 제대로 입력해주세요.");
    }
  }
}
```


[Debug] /날씨 서울
[봇] [서울 지역의 날씨]

상태 : 맑음

현재온도 : 33°C

체감온도 : 35.9°C

미세먼지 : 22 $\mu\text{g}/\text{m}^3$

[Debug] /날씨 춘천

[봇] [춘천 지역의 날씨]

상태 : 구름많음

현재온도 : 32°C

체감온도 : 34.6°C

미세먼지 : 32 $\mu\text{g}/\text{m}^3$

[Debug] /날씨 어딘가

[봇] 에러가 발생했습니다. 지역명을 제대로 입력해주세요.

[Debug] /날씨 뽕

[봇] 에러가 발생했습니다. 지역명을 제대로 입력해주세요.

오케이. 예외처리까지 완벽합니당!

이제 날씨를 Jsoup으로 파싱해봅시다.

(8월 초에 이 글을 쓰기 시작했는데 망할 귀차니즘으로 9월까지 끌고왔..)

이번엔 한번 에버랜드 날씨를 파싱해봅시다(?)

```
var weather = org.jsoup.Jsoup.connect("https://m.search.naver.com/search.naver?query=에버랜드+날씨").get();
```

```
<div class="wt_status">
  <div class="status_bx">
    <div class="status">
      <div class="wt_img">
        <span class="ico_status ws1"><span
class="u_hc">맑음</span></span>
      </div>
      <div class="wt_text">
        <strong><span class="u_hc">현재온도</
span><em class="figure degree_code">17</em></
strong>
        <span class="chill_temp"><span>체감온도</
span><em class="figure degree_code">18.0</em></
span>
      </div>
    </div>
  </div>
```



```

<div class="wt_summary">
  <p class="main">맑음, 어제보다 <em
class="figure degree_code">4</em> 낮아요</p>
  <a nocr onclick="return goOtherCR(this,
'a=nco_x7t*4.pdst&amp;r=1&amp;i=88157038_00000
065035F&amp;u=' + urlencode(this.href));"
href="?where=m&amp;sm=mtb_etc&amp;query=%EA
%B2%BD%EA%B8%B0%EB%8F%84%20%EC%9A%A9
%EC%9D%B8%EC%8B%9C%20%EC%B2%98%EC%9D
%B8%EA%B5%AC%20%ED%8F%AC%EA%B3%A1%EC
%9D%8D%20%EB%AF%B8%EC%84%B8%EB%A8%BC
%EC%A7%80" class="link_dust"> <span>미세먼지</
span><em class="figure sign1">23<span>μg/m³</
span></em> </a>
</div>
<div class="status_list">
  <ul>
    <li><strong>습도 </strong><span

```

저희가 찾으려는 상태, 현재온도, 체감온도, 미세먼지가 다 <div class="status_bx"> 안에 있네요! 이 태그를 select해봅시다.

```

var weather = org.jsoup.Jsoup.connect("https://m.search.naver.com/search.naver?query=에버랜드+날씨").get().select("div.status_bx");

```

```

[봇] <div class="status_bx">
  <div class="status">
    <div class="wt_img">
      <span class="ico_status ws1"><span
class="u_hc">맑음</span></span>
    </div>
    <div class="wt_text">
      <strong><span class="u_hc">현재온도</span><em
class="figure degree_code">17</em></strong>
      <span class="chill_temp"><span>체감온도</
span><em class="figure degree_code">18.0</em></
span>
    </div>
  </div>
  <div class="wt_summary">
    <p class="main">맑음, 어제보다 <em class="figure
degree_code">4</em> 낮아요</p>
    <a nocr onclick="return goOtherCR(this

```

```

<a href="#" onclick="return goodnerok(this,
'a=nco_x7t*4.pdst&r=1&i=88157038_00000
065035F&u=' + urlencode(this.href));"
href="?where=m&sm=mtb_etc&query=%EA
%B2%BD%EA%B8%B0%EB%8F%84%20%EC%9A%A9
%EC%9D%B8%EC%8B%9C%20%EC%B2%98%EC%9D
%B8%EA%B5%AC%20%ED%8F%AC%EA%B3%A1%EC
%9D%8D%20%EB%AF%B8%EC%84%B8%EB%A8%BC
%EC%A7%80" class="link_dust"> <span>미세먼지</
span><em class="figure sign1">23<span>μg/m³</
span></em> </a>
</div>

```

뭔가 복잡해보이지만 차근차근 분석해서 selector들을 만들어봅시다.

일단 상태 '맑음' 은 <div class="wt_img"> 태그 안에 span 태그 안에 span 태그에 들어있군요!

그리고 현재온도의 온도 부분은 <div class="wt_text"> 태그 안 태그 안에 em 이라는 태그명으로 된 태그에 있네요.

체감온도의 온도 부분은 <div class="wt_text"> 안에 span 태그 안에 em 태그에 들어있네요.

마지막으로, 미세먼지는 a태그 안에 들어있는데, a가 상당히 복잡해 보이죠? 별거 없습니다. 링크 때문에 좀 복잡해보였던거였습니다.

아무튼 a에 nocr onclick 말고 class의 속성값을 보면 link_dust라는게 있군요! 그리고 미세먼지의 수치는 em 태그 안에 있네요. 이제 selector를 만들어볼까요?

상태 : div.wt_img > span > span

현재온도 : div.wt_text > strong > em

체감온도 : div.wt_text > span > em

미세먼지 : a.link_dust > em

이제 이걸 바탕으로 최종 소스를 작성해봅시다.

```

var weather = org.jsoup.Jsoup.connect("https://m.search.naver.com/search.naver?query=에버랜드+날씨").get().select("div.status_bx");
var status = weather.select("div.wt_img > span > span").text();
var nowTemp = weather.select("div.wt_text > strong > em").text() + "°C"; //단위 붙여주기
var chillTemp = weather.select("div.wt_text > span > em").text() + "°C"; //단위 붙여주기
var dust = weather.select("a.link_dust > em").text();
replier.reply("[에버랜드 날씨 정보]\n상태 : "+status+"\n현재온도 : "+nowTemp+"\n체감온도 : "+chillTemp+"\n미세먼지 : "+dust);

```

이제 작동시켜 볼까요?

Reply : [에버랜드 날씨 정보]
상태 : 맑음 흐림 흐림 구름많음 흐림
현재온도 : 18 18 27 19 25°C
체감온도 : 18.8°C
미세먼지 : 20μg/m³ 좋음 좋음 좋음 좋음

이런, 상태와 현재온도 그리고 미세먼지 부분의 태그가 한개가 아니었나봅니다. 괜찮아요. get(0)으로 첫번째것만 집어옵시다.

```

var weather = org.jsoup.Jsoup.connect("https://m.search.naver.com/search.naver?query=에버랜드+날씨").get().select("div.status_bx");

```



```

var status = weather.select("div.wt_img > span > span").get(0).text();
var nowTemp = weather.select("div.wt_text > strong > em").get(0).text() + "°C"; //단위 붙여주기
var chillTemp = weather.select("div.wt_text > span > em").text() + "°C"; //단위 붙여주기
var dust = weather.select("a.link_dust > em").get(0).text();
replier.reply("[에버랜드 날씨 정보]\n상태 : "+status+"\n현재온도 : "+nowTemp+"\n체감온도 : "+chillTemp+"\n미세먼지 : "+dust);

```

```

[Debug] var weather =
org.jsoup.Jsoup.connect("https://
m.search.naver.com/
search.naver?query=에버랜드+날씨").get().select("div.s
tatus_bx");
var status = weather.select("div.wt_img > span >
span").get(0).text();
var nowTemp = weather.select("div.wt_text > strong >
em").get(0).text() + "°C"; //단위 붙여주기
var chillTemp = weather.select("div.wt_text > span >
em").text() + "°C"; //단위 붙여주기
var dust = weather.select("a.link_dust >
em").get(0).text();
replier.reply("[에버랜드 날씨 정보]\n상태 :
"+status+"\n현재온도 : "+nowTemp+"\n체감온도 :
"+chillTemp+"\n미세먼지 : "+dust);
[봇] [에버랜드 날씨 정보]
상태 : 맑음
현재온도 : 18°C
체감온도 : 18.8°C
미세먼지 : 20µg/m³

```

완벽...아 참 예외처리를 깜빡했네요! size로 값이 있는지 없는지 체크를 해줘야겠습니다. 동시에 response 함수도 씌워주고요

(실력 상승을 위해 본인이 직접 소스를 짜 봅시다! ^^)

```

[Debug] /날씨 서울
[봇] [서울 날씨 정보]
상태 : 구름많음
현재온도 : 23°C
체감온도 : 24.8°C
미세먼지 : 18µg/m³
[Debug] /날씨 에버랜드
[봇] [에버랜드 날씨 정보]
상태 : 구름많음
현재온도 : 19°C

```

체감온도 : 19.8°C
미세먼지 : 20µg/m³
[Debug] /날씨 환상의 나라
[봇] 해당 지역의 날씨 정보를 불러올 수 없습니다.

Jsoup으로 날씨 파싱도 마무리되었네요!

강좌 끝.

글 중간에 언급했다시피 8월 초에 작성을 시작한 글이 지금까지 미뤄져 왔네요.. 역시 파싱은 재미없..음
다음시간에는 노래가사나 한강 수온을 파싱해볼 생각입니다만 좋은 아이디어가 있다면 그걸 해볼 생각도 있습니당

그럼 이만



마른얼음 BOT님의 게시글 더보기 >

❤ 좋아요 7 💬 댓글 15

🔗 공유 | 신고

댓글 등록순 최신순 C

댓글알림 ☐

삭제된 댓글입니다.



성빈

+1

2019.09.01. 13:26 답글쓰기



마른얼음 BOT 작성자

성빈



2019.09.01. 16:20 답글쓰기



줄려

고인ㅁ...

2019.09.01. 16:19 답글쓰기



마른얼음 BOT 작성자

눅 청정수

2019.09.01. 16:19 답글쓰기



줄려

마른얼음 BOT ::

2019.09.01. 16:21 답글쓰기

삭제된 댓글입니다.



마른얼음 BOT 작성자

이정도면 짧은겁니다(?)

2019.09.02. 17:35 답글쓰기



신광

응 잘 배우고 가네요!

2019.09.02. 17:05 답글 쓰기



마른얼음 BOT 작성자

헬 도움이 되셨다면 감사합니당

2019.09.02. 17:35 답글 쓰기



SP

마른얼음 BOT 넵!

2019.09.02. 18:04 답글 쓰기



Sjy0916

원소 설명 어때요?? 두산백과용

2019.09.07. 20:44 답글 쓰기



마른얼음 BOT 작성자

흐음.. 한번 해보고 귀차니즘이 오면 안하겠습니다(?)

2019.09.07. 20:45 답글 쓰기



Sjy0916

마른얼음 BOT 네 ㅋㅋ 근데 해봤더니 다른데는 다 더럽게 되고 두산백과가 젤 깔끔하네요

2019.09.07. 20:47 답글 쓰기



Sjy0916

Sjy0916 참고로 전 표 부분만 끌어 썼어요

2019.09.07. 20:48 답글 쓰기



Sjy0916

Sjy0916 근데 문제는 원소마다 표에 써있는 설명의 종류가 다 달라요(?) 거기서 막혔네요

2019.09.07. 20:49 답글 쓰기



빛

닥추!!!

2019.09.16. 02:58 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

| | | |
|--|-----------|-------------|
| [중급] (라이노 엔진) 자바스크립트 기초 강의 (5 - 3) - 파싱 예제 두 번째, 노래가사 🧐 [18] | 마른얼음 BOT | 2019.09.07. |
| [기타] Json을 좀 더 편리하게 쓸 수 있게 해주는 툴 2가지 🧐 [2] | Semicolon | 2019.09.05. |
| [중급] (라이노 엔진) 자바스크립트 기초 강의 (5 - 2) - 파싱 예제 첫 번째, 날씨 🧐 [15] | 마른얼음 BOT | 2019.09.01. |
| [초급] 강좌 #4 객체 기본 🧐 [19] | doami2005 | 2019.08.29. |
| [초급] 랜덤 함수 미세먼지 팁 [12] | 사로로 | 2019.08.27. |

이 카페 인기글

[봇드게임] 게이트 v0.9.0

ZUMP

♡0 💬4

요즘 나타나는 새로운 홍보 수법 중 하나가

Dark Tornado

♡0 💬7



태양, 달 정보 구현 완료

중수가 초보에게 하는 강좌 [2]

하하하하

♡1 💬11



기 나무 곡경애(노말)(의
좌악의 문들 중 하나를

5000분의 1을 얻은 사람

부추보안니당

eval 질문

♡0 💬2

Kiri

♡0 💬5



혹시 여기서 변수를 저장하는 부분이 어딘지
아시는분 계시나요..??

특정방에서만 반응하는거..

도배하는 법 (잡글)

Milk2

♡0 💬15

BennyK

♡0 💬5