

| 카카오톡 봇 강좌 | >

[응용] 채팅횟수 및 출석 봇 만들기 강좌

 **OtakoidTony** 채팅 고수 1:1 채팅

2020.07.18. 21:12 조회 109

 댓글 5

 URL 복사



본 강좌는 복불러들을 위한 강좌입니다. 채팅횟수나 출석 봇을 작성해보고 싶으신 분들은 꼭 참고해보세요!

우선 필요한 함수가 4가지가 있다.

1. 채팅횟수에 순위를 매기기 위하여 필요한 객체를 속성을 기준으로 정렬하는 함수.
2. 객체 배열 중에 key에 해당하는 값이 value인 객체의 인덱스를 반환하는 함수.
3. 2번째 함수를 이용해 해당 객체를 내보내는 함수.
4. Date객체에 대하여 같은 날짜인지 아닌지를 반환하는 함수.

우선 첫번째로 채팅횟수에 순위를 매기기 위하여 필요한 객체를 속성을 기준으로 정렬하는 함수는 모질라 사이트에 가면 친절히 쓰여 있다.

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array/sort

물론, 함수로 되어 있지는 않고, 예제로써 작성되어 있다. 이를 참고하여 함수를 작성하면 다음과 같다.

```
/**
 * name을 기준으로 객체배열을 정렬하는 함수
 * @param name 정렬할 기준이 되는 key
 * @param ascending 오름차순 여부
 */
Array.prototype.sort_by = function (name, ascending) {
  if (ascending) {
    return this.sort(function (a, b) {
      if (a[name] > b[name]) {
        return 1;
      }
      if (a[name] < b[name]) {
        return -1;
      }
      return 0;
    });
  } else {
    return this.sort(function (a, b) {
      if (a[name] > b[name]) {
        return -1;
      }
      if (a[name] < b[name]) {
        return 1;
      }
      return 0;
    });
  }
};
```

이제 두번째로 객체 배열 중에 key에 해당하는 값이 value인 객체의 인덱스를 반환하는 함수를 작성하자. 이는 반복문을 이용하면 작성하기 매우 쉽다.

우선 index가 0인 객체부터 index가 배열의 크기-1인 객체까지 확인을 한다. 만약에 반환하고자 하는 객체가 있다면 바로 index를 반환해줌으로써 반복문으로부터 빠져나오도록 한다. 만약에 해당 객체가 존재하지 않는 경우에는 -1을 반환하도록 하자!

작성하면 다음과 같다.

```
/**
 * 객체 배열 중에 key에 해당하는 값이 value인 객체의 인덱스를 반환하는 함수
```

```

* @param key 찾을 값에 대한 key
* @param value key에 해당하는 값
*/
Array.prototype.findObjectIndex = function (key, value) {
    for (var i = 0; i < this.length; i++) if (this[i][key] == value) return i;
    return -1;
};

```

이제 세번째로 2번째 함수를 이용해 해당 객체를 내보내는 함수를 작성해보자. 이는 단순히 if문 하나로 해결되므로 바로 작성한다.

```

/**
 * 객체 배열 중에 key에 해당하는 값이 value인 객체를 반환하는 함수
 * 해당하는 객체가 없을 경우 null을 반환.
 * @param key 찾을 값에 대한 key
 * @param value key에 해당하는 값
 */
Array.prototype.findObject = function (key, value) {
    if (this.findObjectIndex(key, value) != -1) return this[this.findObjectIndex(key, value)];
    else return null;
};

```

뭔가 매우 복잡해보이지만, 다시 한번 유심히 살펴보면 그저 index가 -1일 때는 null을 반환하고 아닌 경우에는 해당 객체를 반환하는 너무 간단한 함수이다.

마지막으로 Date객체에 대하여 같은 날짜인지 아닌지를 반환하는 함수를 작성해보자.

우선 Date객체의 메소드 3가지를 소개해줘야 되겠다.

```

.getDate()
    몇 일인지 반환하는 함수
.getMonth()
    몇 월인지 반환하는 함수
.getFullYear()
    몇 년인지 반환하는 함수

```

이를 참고하면 이 역시 간단하게 작성할 수 있다. 물론 주석에 설명이 들어가 있으니 이해할 수 있을 것이다.

```

/**
 * 입력받은 Date객체에 대하여 같은 날짜인지 아닌지를 반환하는 함수
 * @param targetDate 비교할 Date 객체
 */
Date.prototype.isSameDate = function (targetDate) {
    return (
        /* 일(Day)이 같은지 판별 */
        this.getDate() == targetDate.getDate() &&

        /* 월(Month)이 같은지 판별 */
        this.getMonth() == targetDate.getMonth() &&

        /* 년도(Year)가 같은지 판별 */
        this.getFullYear() == targetDate.getFullYear()
    );
};

```

이걸로 필요한 함수들은 전부 작성하였다. 이제 이들을 이용하여 채팅 횟수 및 출석 봇을 작성해볼 것이다.

* * * * *

우선, 봇을 켜다 켜을 때, 그동안의 데이터가 전부 증발되면 안되니까, FileStream을 이용할 것이다. 물론 FileStream에 대해서 사용 하는 법을 모를 수도 있겠다. 간단히 설명하자면 다음과 같다.

```

FileStream.read(String)
    String(파일 경로)에 파일이 실제로 존재하면 해당 파일 내 텍스트 내용을 내보내고 존재하지 않을 시 null 값을 내보낸다.

```

```

FileStream.write(String, String)
    첫번째 인자(파일 경로)라는 파일에 두번째 인자(텍스트 내용)을 저장한다. 만약 파일이 없으면 만들어내서라도 저장한다.

```

저자는 sdcard/Kakao_senderData/senderData.json이라는 파일에 데이터를 저장할 것이며, 우선은 다음과 같이 작성한다.

```
var senderData = FileStream.read("sdcard/Kakao_senderData/senderData.json");
if (senderData == null) senderData = {};
else senderData = JSON.parse(senderData);
```

senderData에 파일을 읽고 객체화 시킨다. JSON.parse()는 문자열을 객체로 만들어 주는 역할을 한다.

이제 본격적으로 response()를 작성할 것이다.

우선, senderData에 어떤 객체를 넣으면 좋을까?

저자는 다음과 같은 구조를 갖는 객체를 이용하기로 하였다.

```
{
  채팅방_이름: [
    {
      'name' : 유저_닉네임,
      'score' : 채팅횟수,
      'time' : Date객체
    }
  ]
}
```

뭐 이 정도로만 설명해도 될까라고 생각한다. 물론 이대로 쓰는 거는 아니다. 위 코드는 단순히 예시로 틀만 보여주기 위함이다.

우선 답장이 오면 다음 루틴을 따른다. 여기서부터는 function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId) { } 내부에 작성한다.

먼저 senderData에 room이 있는지를 확인하고 없는 경우네는 senderData[room] = [];를 해줌으로써 senderData에 room에 대한 배열을 추가해준다.

이는 다음과 같이 작성한다.

```
if (!(room in senderData)) {
  senderData[room] = [];
}
```

그런 다음, 이제 해당 room에 대하여 'name'이 sender인 객체가 있는지를 확인하고 없으면 해당 데이터를 기록하고, 있는 경우에는 해당 객체에 대하여 'score'에는 1 증가시키고 'time'에는 Date 객체를 할당해준다.

이는 다음과 같이 작성한다.

```
if (senderData[room].findObjectIndex('name', sender) == -1) {
  senderData[room].push({
    'name': sender,
    'score': 1,
    'time': (new Date()).toString()
  });
} else {
  senderData[room].findObject('name', sender)['score'] += 1;
  senderData[room].findObject('name', sender)['time'] = (new Date()).toString();
}
```

이걸로 채팅횟수 및 출석에 대한 데이터는 모두 저장하였다.

이제 이걸 출력하는 기능을 넣으면 끝난다.

우선, 랭킹에 대해 출력해보자. 이는 먼저 senderData[room].sort_by('score', ascending = false);를 해줌으로써 senderData[room]을 score 속성을 기준으로 정렬한 다음, 출력한다. 이는 다음과 같이 작성하면 된다.

```
if (msg == "!랭킹") {
  senderData[room].sort_by('score', ascending = false);
  var output = '\u200b'.repeat(500) + '\n\n';
  for (var i = 0; i < senderData[room].length; i++) {
    user_time = new Date(senderData[room][i].time);
```

```

        user_time = new Date(senderData[room][i].time);
        output += 'Name: ' + senderData[room][i].name + '\n';
        output += 'Rank: ' + (i + 1) + "등" + '\n';
        output += 'Score: ' + senderData[room][i].score + '\n';
        if (i == senderData[room].length - 1) {
            output += 'Time: ' + user_time.getFullYear() + "년 " + (user_time.getMonth() + 1) + "월 " + user_time.
getDate() + "일";
        } else {
            output += 'Time: ' + user_time.getFullYear() + "년 " + (user_time.getMonth() + 1) + "월 " + user_time.
getDate() + "일" + '\n\n';
        }
    }
    replier.reply(output);
}

```

그리고 마지막으로 출석부를 출력해보자. 먼저 메시지를 보낸 날짜와 비교를 해야 하므로 Date 객체를 하나 만들어 준다. 그런 다음, for 문을 이용해 첫번째 객체부터 마지막 객체까지 배열을 순회해가면서 if문을 이용해 .isSameDate(만들었던 Date 객체)의 값이 참인 경우만 출력을 해준다. 이는 다음과 같이 작성할 수 있다.

```

if (msg == "!출석부") {
    senderData[room].sort_by('score', ascending = false);
    var output = '\u200b'.repeat(500) + '\n\n';
    var today = new Date();
    for (var i = 0; i < senderData[room].length; i++) {
        user_time = new Date(senderData[room][i].time);
        if (user_time.isSameDate(today)) {
            if (i == senderData[room].length - 1) {
                output += 'Name: ' + senderData[room][i].name;
            } else {
                output += 'Name: ' + senderData[room][i].name + '\n\n';
            }
        }
    }
    replier.reply(output);
}

```

마지막으로, 데이터가 증발하면 안되므로 다음 한 줄을 작성한다.

```

FileStream.write("sdcard/Kakao_senderData/senderData.json", JSON.stringify(senderData, null, '\t'));

```

이걸로 채팅횟수 및 출석 봇 만들기 강좌를 마친다.

좀더 기능이 많은 코드는 <https://github.com/OtakoidTony/MessengerBotExample/blob/master/KakaoSenderData.js> 를 참고하면 된다.

終



OtakoidTony님의 게시글 더보기 >

❤️ 좋아요 1 💬 댓글 5

🔗 공유 | 신고

댓글 등록순 최신순

댓글알림 ☐



Hibot

임시저장있어욤!

2020.07.18. 21:15 답글쓰기



OtakoidTony 작성자

임시저장 기능을 안써봐서;;; ㅎㅎ;;

2020.07.18. 21:23 답글쓰기



김애용

굿~~~~~

2020.07.18. 21:21 답글쓰기



성빈
아하!

2020.07.18. 21:34 답글 쓰기



고기만두
다음에는 가르치기 강좌도 부탁드립니다!

2020.07.18. 23:25 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'카카오톡 봇 강좌' 게시판 글

전체 [응용] 말머리 글

이 게시판 새글 구독하기 ☐

[응용] 가르치기 봇 만들기 강좌 [9] N	OtakoidTony	00:52
[응용] 채팅횟수 및 출석 봇 만들기 강좌 [5] N	OtakoidTony	2020.07.18.
[초급] 기본 개념에 대한 강좌 조건문과 반복문 [6] N	OtakoidTony	2020.07.18.
[중급] 네이버에서 지역별 날씨 정보 크롤링하기 / 스압 알아서 주의 🙏📷 [7]	Dark Tornado	2020.07.15.
함수지향과 객체지향(17) - 참조를 알아보자 [6]	choi0108	2020.07.12.

1 2 3

전체보기

이 카페 인기글



메신저봇 글씨 색

Lunar
♡0 💬6

버튼 누르면 intent가 함수를 실행하게하려
면

jomin398
♡0 💬20

새로 왔습니다

ArtistBRIN
♡0 💬7



아키네이터

실시간 멜론 차트

여러분은 어캐 카톡봇 배우게 되었나요?

사로로

♡0 💬25

채팅횟수 및 출석 봇 만들기 강좌

OtakoidTony

♡1 💬5

getSchoolFood

도미 doami2005

♡0 💬6



웹툰