



스파르타코딩클럽 7기 3주차 - 학습자료

[수업 목표]

1. Javascript + jQuery + Ajax를 활용하는 기본 패턴을 익힌다.
2. 파이썬 기초 문법을 안다.
3. 원하는 페이지를 크롤링 할 수 있다.

🔊: 실습 시간

* 강의 상황에 따라, 시간은 유동적일 수 있습니다.

전반 3시간

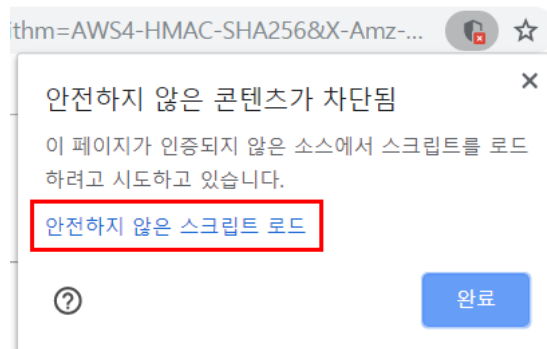
[시작]

- ▼ "15초 체크인"을 진행합니다.
 - 튜터는 타이머를 띄워주세요! ([링크](#)).
 - 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.
- ▼ 간단 복습하기 & 오늘 배울 것 이야기
 - 튜터님이 5분 동안 지난번에 배웠던 키워드들과, 오늘 배울것을 이야기합니다.

[2시간]: Javascript + jQuery + Ajax 연습하기

- ▼ 1) 서울시 OpenAPI 활용하기
 - ▼ (0) 완성본을 보는 방법

👉 https로 띄운 페이지에서, http에 Ajax요청을 하기 때문에 경고창을 띄웁니다.
상관 없도록, "안전하지 않은 스크립트 로드"를 클릭해야 제대로 보입니다!



- ▼ (1) 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기



사용 할 API :

<http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99>

관련 웹사이트: <https://data.seoul.go.kr/dataList/datasetList.do>

? 잠시 Review quiz: API 주소 마지막 숫자를 99 → 2로 바꾸면 무슨 일이 벌어질까요? 이 방식은 GET/POST 중 어느 방식일까요?

▼ 퀴즈 코드(빈칸)

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>jQuery 연습하고 가기!</title>

<!-- JQuery를 import 합니다 -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<style type="text/css">
    div.question-box {
        margin: 10px 0 20px 0;
    }
</style>

<script>
    function q1() {
        // 여기에 코드를 입력하세요
    }
</script>

</head>
<body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
        <h2>1. 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기</h2>
        <p>모든 구의 미세먼지를 표기해주세요</p>
        <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌어져야 합니다.</p>
        <button onclick="q1()">업데이트</button>
        <ul id="names-q1">
            <li>종로구 : 82</li>
            <li>종로구 : 87</li>
            <li>용산구 : 84</li>
            <li>은평구 : 82</li>
        </ul>
    </div>
</body>
</html>
```

• 완성본 보기 (링크)

▼ 완성본 코드

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>jQuery 연습하고 가기!</title>

<!-- JQuery를 import 합니다 -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<style type="text/css">
    div.question-box {
        margin: 10px 0 20px 0;
    }
</style>

<script>
    function q1() {
        function q1() {
            $('#names-q1').html('');
            $.ajax({
                type: "GET",
```

```

        url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
        data: {},
        success: function(response){
            let rows = response["RealtimeCityAir"]["row"];
            for (let i = 0; i < rows.length; i++) {
                let gu_name = rows[i]['MSRSTE_NM'];
                let gu_mise = rows[i]['IDEX_MVL'];
                let temp_html = '<li>'+gu_name+' : '+gu_mise+'</li>'
                $('#names-q1').append(temp_html);
            }
        }
    })
}

</script>

</head>
<body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
        <h2>1. 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기</h2>
        <p>모든 구의 미세먼지를 표기해주세요</p>
        <p>업데이트 버튼을 누를 때마다 지웠다 새로 써여져야 합니다.</p>
        <button onclick="q1()">업데이트</button>
        <ul id="names-q1">
        </ul>
    </div>
</body>
</html>

```

▼ (2) 서울시 OpenAPI(실시간 따릉이 현황)을 이용하기



사용 할 API :

<http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99>

관련 웹사이트: <https://data.seoul.go.kr/dataList/datasetList.do>

▼ 퀴즈 코드(빈칸)

```

<!doctype html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>jQuery 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
        div.question-box {
            margin: 10px 0 20px 0;
        }
        table {
            border: 1px solid;
            border-collapse: collapse;
        }

        td, th {
            padding: 10px;
            border: 1px solid;
        }
    </style>

    <script>
        function q1() {
            // 여기에 코드를 입력하세요
        }
    </script>

</head>
<body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">

```

```

<h2>2. 서울시 OpenAPI(실시간 따릉기 현황)를 이용하기</h2>
<p>모든 위치의 따릉기 현황을 보여주세요</p>
<p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
<button onclick="q1()">업데이트</button>
<table>
  <thead>
    <tr>
      <td>거치대 위치</td>
      <td>거치대 수</td>
      <td>현재 거치된 따릉이 수</td>
    </tr>
  </thead>
  <tbody id="names-q1">
  </tbody>
</table>
</div>
</body>
</html>

```

- [완성본 보기 \(링크\)](#)

▼ 완성본 코드

```

<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>jQuery 연습하고 가기!</title>
  <!-- JQuery를 import 합니다 -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }
    table {
      border: 1px solid;
      border-collapse: collapse;
    }

    td, th {
      padding: 10px;
      border: 1px solid;
    }
  </style>

  <script>
    function q1() {
      $('#names-q1').html('');
      $.ajax({
        type: "GET",
        url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99",
        data: {},
        success: function(response){
          let rows = response["rentBikeStatus"]["row"];
          for (let i = 0; i < rows.length; i++) {
            let rack_name = rows[i]['stationName'];
            let rack_cnt = rows[i]['rackTotCnt'];
            let bike_cnt = rows[i]['parkingBikeTotCnt'];
            let temp_html = '<tr>\n
                                <td>' + rack_name + '</td>\n
                                <td>' + rack_cnt + '</td>\n
                                <td>' + bike_cnt + '</td>\n
                                </tr>'
            $('#names-q1').append(temp_html);
          }
        }
      })
    }
  </script>

</head>
<body>
  <h1>jQuery+Ajax의 조합을 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>2. 서울시 OpenAPI(실시간 따릉기 현황)를 이용하기</h2>
    <p>모든 위치의 따릉기 현황을 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <table>
      <thead>

```

```

        <tr>
            <td>거치대 위치</td>
            <td>거치대 수</td>
            <td>현재 거치된 따릉이 수</td>
        </tr>
    </thead>
    <tbody id="names-q1">
    </tbody>
</table>
</div>
</body>
</html>

```



[한걸음 더]

따릉이 대수가 5대 미만인 곳은 빨강게 보여주면 어떨까요?

- [완성본 보기 \(링크\)](#).

▼ 완성본 코드

```

<!doctype html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>jQuery 연습하고 가자!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
        div.question-box {
            margin: 10px 0 20px 0;
        }
        table {
            border: 1px solid;
            border-collapse: collapse;
        }

        td, th {
            padding: 10px;
            border: 1px solid;
        }
        .urgent {
            color: red;
            font-weight: bold;
        }
    </style>

    <script>
    function q1() {
        $('#names-q1').html('');
        $.ajax({
            type: "GET",
            url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99",
            data: {},
            success: function(response){
                let rows = response["rentBikeStatus"]["row"];
                for (let i = 0; i < rows.length; i++) {
                    let rack_name = rows[i]['stationName'];
                    let rack_cnt = rows[i]['rackTotCnt'];
                    let bike_cnt = rows[i]['parkingBikeTotCnt'];
                    let temp_html = '';
                    if (bike_cnt < 5) {
                        temp_html = '<tr class="urgent">\
                            <td>' + rack_name + '</td>\
                            <td>' + rack_cnt + '</td>\
                            <td>' + bike_cnt + '</td>\
                        </tr>'
                    } else {
                        temp_html = '<tr>\
                            <td>' + rack_name + '</td>\
                            <td>' + rack_cnt + '</td>\
                            <td>' + bike_cnt + '</td>\
                        </tr>'
                    }
                    $('#names-q1').append(temp_html);
                }
            }
        })
    }
    </script>

```

```

</script>

</head>
<body>
  <h1>jQuery+Ajax의 조합을 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>2. 서울시 OpenAPI(실시간 따릉기 현황)를 이용하기</h2>
    <p>모든 위치의 따릉기 현황을 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <table>
      <thead>
        <tr>
          <td>거치대 위치</td>
          <td>거치대 수</td>
          <td>현재 거치된 따릉기 수</td>
        </tr>
      </thead>
      <tbody id="names-q1">
      </tbody>
    </table>
  </div>
</body>
</html>

```

▼ 2) 일반 API 활용하기

▼ (1) 랜덤 숫자 의미부여 API를 이용하기(각 숫자의 의미를 알려주는 API!)

👉 사용 할 API : <http://numbersapi.com/숫자>

예) <http://numbersapi.com/7000>
<http://numbersapi.com/1>

관련 웹사이트 : <http://numbersapi.com/>

▼ 퀴즈 코드(빈칸)

```

<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>jQuery 연습하고 가자!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
      div.question-box {
        margin: 10px 0 20px 0;
      }
    </style>

    <script>
      function q1() {
        // 여기에 코드를 입력하세요
      }
    </script>

  </head>
  <body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
      <h2>4. 랜덤 숫자 의미부여 API를 이용하기</h2>
      <p>모든 숫자는 어떤 의미가 있습니다.</p>
      <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
      <input type="number" id="magicnum">
      <button onclick="q1()">이 숫자의 의미는?</button>
      <h5 id="meaning"></h5>
    </div>
  </body>
</html>

```

- [완성본 보기 \(링크\)](#)

▼ 완성본 코드

```
<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>JQuery 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
      div.question-box {
        margin: 10px 0 20px 0;
      }
    </style>

    <script>
      function q1() {
        let target_num = $('#magicnum').val();

        if (target_num == '') {
          alert('숫자를 입력하세요');
          return;
        }

        let target_url = 'http://numbersapi.com/'+target_num;
        $.ajax({
          type: "GET",
          url: target_url,
          data: {},
          success: function(response){
            $('#meaning').text(response);
          }
        })
      }
    </script>

  </head>
  <body>
    <h1>JQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
      <h2>4. 랜덤 숫자 의미부여 API를 이용하기</h2>
      <p>모든 숫자는 어떤 의미가 있습니다. 함께 살펴볼까요?</p>
      <p>아무것도 입력하지 않으면, alert을 띄워주세요.</p>
      <input type="number" id="magicnum">
      <button onclick="q1()">이 숫자의 의미는?</button>
      <h5 id="meaning"></h5>
    </div>
  </body>
</html>
```

▼ (2) 랜덤 고양이 사진 API를 이용하기

 사용 할 API : <https://api.thecatapi.com/v1/images/search>

관련 웹사이트: <https://docs.thecatapi.com/>

▼ 퀴즈 코드(빈칸)

```
<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>JQuery 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
      div.question-box {
        margin: 10px 0 20px 0;
      }
      div.question-box > div {
        margin-top: 30px;
      }
    </style>

    <script>
      function q1() {
        let target_num = $('#magicnum').val();

        if (target_num == '') {
          alert('숫자를 입력하세요');
          return;
        }

        let target_url = 'http://numbersapi.com/'+target_num;
        $.ajax({
          type: "GET",
          url: target_url,
          data: {},
          success: function(response){
            $('#meaning').text(response);
          }
        })
      }
    </script>

  </head>
  <body>
    <h1>JQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
      <h2>4. 랜덤 숫자 의미부여 API를 이용하기</h2>
      <p>모든 숫자는 어떤 의미가 있습니다. 함께 살펴볼까요?</p>
      <p>아무것도 입력하지 않으면, alert을 띄워주세요.</p>
      <input type="number" id="magicnum">
      <button onclick="q1()">이 숫자의 의미는?</button>
      <h5 id="meaning"></h5>
    </div>
  </body>
</html>
```

```

    }

</style>

<script>
    function q1() {
        // 여기에 코드를 입력하세요
    }
</script>

</head>
<body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>

    <div class="question-box">
        <h2>3. 랜덤 고양이 사진 API를 이용하기</h2>
        <p>예쁜 고양이 사진을 보여주세요</p>
        <p>업데이트 버튼을 누를 때마다 지웠다 새로 써여져야 합니다.</p>
        <button onclick="q1()">고양이를 보자</button>
    <div>
        
    </div>
    </div>
</body>
</html>

```

- [완성본 보기 \(링크\)](#)

▼ 완성본 코드

```

<!doctype html>
<html lang="ko">
    <head>
        <meta charset="UTF-8">
        <title>jQuery 연습하고 가기!</title>
        <!-- JQuery를 import 합니다 -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

        <style type="text/css">
            div.question-box {
                margin: 10px 0 20px 0;
            }
            div.question-box > div {
                margin-top: 30px;
            }
        </style>

        <script>
            function q1() {
                $.ajax({
                    type: "GET",
                    url: "https://api.thecatapi.com/v1/images/search",
                    data: {},
                    success: function(response){
                        let imgurl = response[0]['url'];
                        $("#img-cat").attr("src", imgurl);
                    }
                })
            }
        </script>

    </head>
    <body>
        <h1>jQuery+Ajax의 조합을 연습하자!</h1>

        <hr/>

        <div class="question-box">
            <h2>3. 랜덤 고양이 사진 API를 이용하기</h2>
            <p>예쁜 고양이 사진을 보여주세요</p>
            <p>업데이트 버튼을 누를 때마다 지웠다 새로 써여져야 합니다.</p>
            <button onclick="q1()">고양이를 보자</button>
        <div>
            
        </div>
    </div>
</body>
</html>

```


▼ 3) 뉴스 API 활용하기

▼ (1) 뉴스 검색 API를 이용하기



사용 할 API : <https://newsapi.org/v2/everything?q=검색어&apiKey=API키>

예) <https://newsapi.org/v2/everything?q=kpop&apiKey=5d6c1a1d4c0d4de1a60bc34087bb8655>
(q=아이폰)과 같은 한글도 됩니다

관련 웹사이트: <https://newsapi.org/docs/endpoints/everything>

▼ 퀴즈 코드(빈칸)

```
<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>jQuery 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
      div.question-box {
        margin: 10px 0 20px 0;
      }
      div.newsbox {
        width: 500px;
      }

      div.newsbox > div {
        margin-top: 50px;
        border: 1.5px solid #181818;
        padding: 10px;
      }

      div.newsbox > div > img {
        display: block;
      }

      div.newsbox > div > div.title-box {
        margin: 30px 0 30px 0;
      }

      div.newsbox > div > div > a {
        color: #181818;
        font-size: 16px;
        font-weight: bold;
      }

      div.newsbox > div > div > h5 {
        color: #5d5d5d;
        margin: 4px 0 2px 0;
      }

      div.newsbox > div > p {
        background-color: #d3e1e5;
        padding: 5px;
      }
    </style>

    <script>
      function q1() {
        let keyword = $('#input-keyword').val();
        if (keyword == '') {
          alert('키워드를 입력하세요!')
          return;
        }

        // 여기에 코드를 입력하세요

      }
    </script>
  </head>
  <body>
    <h1>jQuery+Ajax의 조합을 연습하자!</h1>

    <hr/>
```

```

<div class="question-box">
  <h2>5. 뉴스 검색 API를 이용하기</h2>
  <p>키워드에 걸맞는 뉴스를 띄워줍니다.</p>
  <p>아무것도 입력하지 않으면, alert을 띄워주세요.</p>
  <input type="text" id="input-keyword" placeholder="검색어를 입력하세요">
  <button onclick="q1()">관련 기사보기</button>
</div>
<hr/>
<div class="newsbox" id="news">
  <div>
    
    <div class="title-box">
      <a href="https://naver.com/" target="_blank">11번가, 애플뮤직 7개월 무료 이용 혜택 제공 나서</a>
    </div>
    <h5>2019-10-29T17:15:00Z | 글쓴이 </h5>
  </div>
  <p>여기에 기사 내용이 들어갑니다. 이 칸을 복사해서 JQuery로 html append 하는 데 쓰면 되겠죠? CSS는 추가/수정 할 필요 없습니다!</p>
</div>
</body>
</html>

```

- [완성본 보기 \(링크\)](#).

▼ 완성본 코드

```

<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <title>JQuery 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
      div.question-box {
        margin: 10px 0 20px 0;
      }
      div.newsbox {
        width: 500px;
      }

      div.newsbox > div {
        margin-top: 50px;
        border: 1.5px solid #181818;
        padding: 10px;
      }

      div.newsbox > div > img {
        display: block;
      }

      div.newsbox > div > div.title-box {
        margin: 30px 0 30px 0;
      }

      div.newsbox > div > div > a {
        color: #181818;
        font-size: 16px;
        font-weight: bold;
      }

      div.newsbox > div > div > h5 {
        color: #5d5d5d;
        margin: 4px 0 2px 0;
      }

      div.newsbox > div > p {
        background-color: #d3e1e5;
        padding: 5px;
      }
    </style>

    <script>
      function q1() {
        let keyword = $('#input-keyword').val();
        if (keyword == '') {
          alert('키워드를 입력하세요!')
          return;
        }

        $('#news').html('');

        let target_url = 'https://newsapi.org/v2/everything?q='+keyword+'&apiKey=5d6c1a1d4c0d4de1a60bc34087bb8655';

```

```

$.ajax({
  type: "GET",
  url: target_url,
  data: {},
  success: function(response){
    let articles = response['articles'];
    for (let i = 0 ; i < articles.length; i++){
      let url = articles[i]['url'];
      let title = articles[i]['title'];
      let imgurl = articles[i]['urlToImage'];
      let date = articles[i]['publishedAt'];
      let author = articles[i]['author'];
      let content = articles[i]['content'];

      let temp_html = '<div>\
        \
        <div class="title-box">\
        <a href="'+url+'" target="_blank">'+title+'</a>\
        <h5>'+date+' | '+author+'</h5>\
        </div>\
        <p>'+content+'</p>\
        </div>';

      $('#news').append(temp_html);
    }
  }
})

</script>

</head>
<body>
  <h1>jQuery+Ajax의 조합을 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>5. 뉴스 검색 API를 이용하기</h2>
    <p>키워드에 걸맞는 뉴스를 띄워줍니다.</p>
    <p>아무것도 입력하지 않으면, alert을 띄워주세요.</p>
    <input type="text" id="input-keyword" placeholder="검색어를 입력하세요">
    <button onclick="q1()">관련 기사보기</button>
  </div>
  <hr/>
  <div class="newsbox" id="news">
  </div>
</body>
</html>

```

[1시간]: "나홀로메모장"에 OpenAPI 붙여보기

▼ 4) "나홀로메모장"에 사용할 수 있도록, OpenAPI를 하나 만들어 두었습니다.

- 완성본을 다시 보고, 생각해봅시다!

👉 링크 들어가보기: <http://spartacodingclub.shop/>

- 아래 API는 무엇을 의미할까요?

👉 링크 들어가보기: <http://spartacodingclub.shop/post>

👉 네 그렇습니다! 나홀로메모장에 들어가는 아티클들의 정보를 불러오는 OpenAPI입니다. 이 API를 써서 저장된 포스팅 불러오기를 만들어볼게요!
(나중에 이 API를 실제로 만들어 볼 예정!)

▼ 5) 포스팅가져오기API 붙이기

1. 1주차에 완성했던 '나홀로메모장'을 켜봅시다!

▼ 다음 코드에서 시작하세요!

```

<!doctype html>
<html lang="en">

<head>

  <!-- Webpage Title -->
  <title>Hello, world!</title>

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
    integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

  <!-- JS -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
    integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
    crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
    integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
    crossorigin="anonymous"></script>

  <!-- 구글폰트 -->
  <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

  <!-- style -->
  <style type="text/css">
    * {
      font-family: 'Stylish', sans-serif;
    }

    .wrap {
      width: 900px;
      margin: auto;
    }

    .comment {
      color: blue;
      font-weight: bold;
    }

    .form-post {
      max-width: 500px;
      padding: 2rem;
      margin: 2rem auto;
      border-color: #e9ecef;
      border-radius: 0.3rem;
      border: solid;
      display: block;
    }

    #post-box {
      display: none;
    }
  </style>

  <script>
    function openclose() {
      // id 값 post-box의 display 값이 block 이면
      if ($('#post-box').css('display') == 'block') {
        // post-box를 가리고
        $('#post-box').hide();
        $('#btn-posting-box').text('포스팅 박스 열기');
      } else {
        // 아니면 post-box를 펴라
        $('#post-box').show();
        $('#btn-posting-box').text('포스팅 박스 닫기');
      }
    }
  </script>

</head>

<body>
  <div class="wrap">
    <div class="jumbotron">
      <h1 class="display-4">나홀로 링크 메모장!</h1>
      <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
      <hr class="my-4">
      <p class="lead">
        <button id="btn-posting-box" onclick="openclose()" type="button" class="btn btn-primary">포스팅 박스 열기</button>
      </p>
    </div>
  </div>

```

```

        </p>
    </div>
    <div class="form-post" id="post-box">
        <div>
            <div class="form-group">
                <label for="exampleFormControlInput1">아티클 URL</label>
                <input class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="exampleFormControlTextarea1">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
            <button type="button" class="btn btn-primary">기사저장</button>
        </div>
    </div>
    <div class="card-columns" id="cards-box">
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
    </div>
</div>
</body>
</html>

```

2. 우선, 로드가 다 되면 실행되는 함수를 하나 만듭니다

👉 구글링해보기: javascript 로딩 후 실행

```
$(document).ready(function(){
    listing();
});

function listing() {
    console.log('화면 로딩 후 잘 실행되었습니다');
}
```

3. API 결과값을 다시한번 확인하기

```
$(document).ready(function(){
    listing();
});

function listing() {
    $.ajax({
        type: "GET",
        url: "http://spartacodingclub.shop/post",
        data: {},
        success: function(response){
            console.log(response)
        }
    })
}
```

👉 response['result'] 에 정상 여부가 있고, response['articles']에 리스트로 아티클들이 들어가 있습니다.

4. 기사데이터를 log에 찍어봅시다!

```
$.ajax({
    type: "GET",
    url: "http://spartacodingclub.shop/post",
    data: {},
    success: function(response){
        console.log(response['articles'])
    }
})
```

5. articles를 돌면서, 하나씩 출력해봅시다.

```
$.ajax({
    type: "GET",
    url: "http://spartacodingclub.shop/post",
    data: {},
    success: function(response){
        let articles = response['articles'];
        for (let i = 0; i < articles.length; i++) {
            console.log(articles[i]);
        }
    }
})
```

6. article 내용(comment, desc, image, title, url)으로 카드를 만들어주는 함수를 만들어봅시다!

```
function make_card(comment, desc, image, title, url) {
    let temp_html = '<div class="card">\n
    \n
    <div class="card-body">\n
    <a href="'+url+'" target="_blank" class="card-title">'+title+'</a>\n
    <p class="card-text">'+desc+'</p>\n
    <p class="card-text comment">'+comment+'</p>\n
    </div>\n
    </div>';
    $('#cards-box').append(temp_html);
}
```

7. make_card 함수를 ajax에 연결합니다.

```
$.ajax({
  type: "GET",
  url: "http://spartacodingclub.shop/post",
  data: {},
  success: function(response){
    let articles = response['articles'];
    for (let i = 0; i < articles.length; i++) {
      make_card(articles[i]['comment'],articles[i]['desc'],articles[i]['image'],articles[i]['title'],articles[i]['url'])
    }
  }
})
```

8. 먼저 있던 카드들을 지워줍니다.

```
$(document).ready(function(){
  $('#cards-box').html('');
  listing();
});
```

9. 1~7 완성된 코드

```
$(document).ready(function(){
  $('#cards-box').html('');
  listing();
});

function listing() {
  $.ajax({
    type: "GET",
    url: "http://spartacodingclub.shop/post",
    data: {},
    success: function(response){
      let articles = response['articles'];
      for (let i = 0; i < articles.length; i++) {
        make_card(articles[i]['comment'],articles[i]['desc'],articles[i]['image'],articles[i]['title'],articles[i]['url'])
      }
    }
  })
}

function make_card(comment, desc, image, title, url) {
  let temp_html = '<div class="card">\n
  \n
  <div class="card-body">\n
  <a href="'+url+'" class="card-title">'+title+'</a>\n
  <p class="card-text">'+desc+'</p>\n
  <p class="card-text comment">'+comment+'</p>\n
  </div>\n
  </div>';
  $('#cards-box').append(temp_html);
}
```

10. 완성된 코드 전체

▼ 코드

```
<!doctype html>
<html lang="en">
<head>

  <!-- Webpage Title -->
  <title>Hello, world!</title>

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-IqfP6Ib5Jn0205Q1bR517-A77aQ14I1J5bzq4a1i0785fcb612320016" crossorigin="anonymous">

  <!-- JS -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbghs" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02" crossorigin="anonymous"></script>
```

```

<!-- 구글폰트 -->
<link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

<script>
function openclose() {
  if ( $('#posting-box').css('display') == 'block' ) {
    $('#posting-box').hide();
    $('#btn-posting-box').text('포스팅 박스 열기')
  } else {
    $('#posting-box').show();
    $('#btn-posting-box').text('포스팅 박스 닫기')
  }
}

$(document).ready(function(){
  $('#cards-box').html('');
  listing();
});

function listing() {
  $.ajax({
    type: "GET",
    url: "http://spartacodingclub.shop/post",
    data: {},
    success: function(response){
      let articles = response['articles'];
      for (let i = 0; i < articles.length; i++) {
        make_card(articles[i]['comment'],articles[i]['desc'],articles[i]['image'],articles[i]['title'],articles[i]['url'])
      }
    }
  })
}

function make_card(comment, desc, image, title, url) {
  let temp_html = '<div class="card">\
    \
    <div class="card-body">\
      <a href="'+url+'" class="card-title">'+title+'</a>\
      <p class="card-text">'+desc+'</p>\
      <p class="card-text comment">'+comment+'</p>\
    </div>\
  </div>';
  $('#cards-box').append(temp_html);
}
</script>

<!-- style -->
<style type="text/css">
* {
  font-family: 'Stylish', sans-serif;
}
.wrap {
  width: 900px;
  margin: auto;
}
.comment {
  color: blue;
  font-weight: bold;
}
.form-post {
  max-width: 500px;
  padding: 2rem;
  margin: 2rem auto;
  border-color: #e9ecef;
  border-radius: 0.3rem;
  border: solid;
  display: block;
}
#posting-box {
  display: none;
}
</style>

</head>
<body>
<div class="wrap">
  <div class="jumbotron">
    <h1 class="display-4">나홀로 링크 메모장!</h1>
    <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
    <hr class="my-4">
    <p class="lead">
      <a id="btn-posting-box" onclick="openclose()" class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
    </p>
  </div>
  <div class="form-post" id="posting-box">
    <div>
      <div class="form-group">

```



```
<label for="exampleFormControlInput1">아티클 URL</label>  
<input id="posting-url" class="form-control" placeholder="">  
</div>  
<div class="form-group">  
  <label for="exampleFormControlTextarea1">간단 코멘트</label>  
  <textarea id="posting-comment" class="form-control" rows="2"></textarea>  
</div>  
<button onclick="posting()" class="btn btn-primary">기사저장</button>  
</div>  
</div>  
<div class="card-columns" id="cards-box">  
  <div class="card">  
      
    <div class="card-body">  
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
  <div class="card">  
      
    <div class="card-body">  
      <h5 class="card-title">여기 기사 제목이 들어가죠</h5>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
  <div class="card">  
      
    <div class="card-body">  
      <h5 class="card-title">여기 기사 제목이 들어가죠</h5>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
  <div class="card">  
      
    <div class="card-body">  
      <h5 class="card-title">여기 기사 제목이 들어가죠</h5>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
  <div class="card">  
      
    <div class="card-body">  
      <h5 class="card-title">여기 기사 제목이 들어가죠</h5>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
  <div class="card">  
      
    <div class="card-body">  
      <h5 class="card-title">여기 기사 제목이 들어가죠</h5>  
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산  
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>  
    </div>  
  </div>  
</div>  
</div>  
</body>  
</html>
```

후반 3시간



[시작하기 전에]

pycharm에서 **new project**를 클릭해서, 수업 폴더 내에 프로젝트를 만들고 시작할게요!

(파일이 섞이지 않도록! 이름은 자유)

[1시간] : 파이썬 기초 문법 익히기

- ▼ 6) 파이썬을 설치한다는 것의 의미 / 파이참은 무엇일까?
- 파이썬을 설치한다?



파이썬을 설치한다는 것: 일종의 번역팩을 설치한다고 생각하면 됩니다. 컴퓨터는 101010001 과 같은 언어만 알아듣는다고 했지요? 파이썬 문법으로 된 것을 101010001로 변환해줄 수 있도록, 번역 패키지를 설치하는 것입니다.

- 파이썬은 무엇일까?



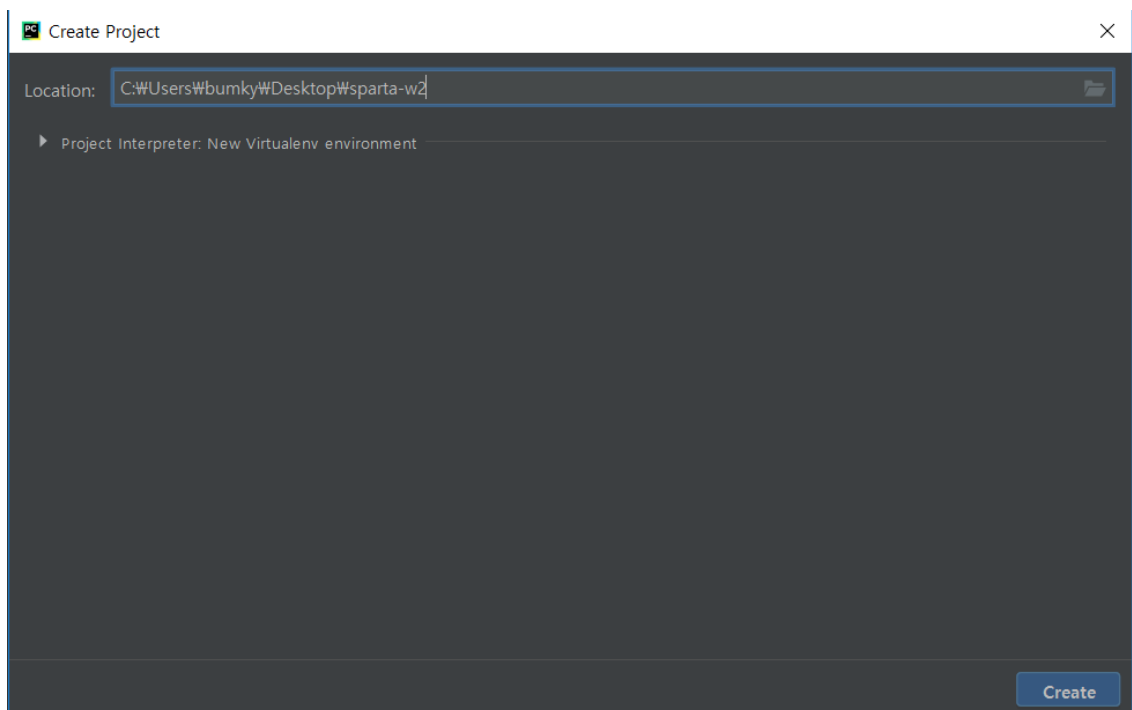
PyCharm은 파이썬을 이용해 코딩하는데 유용한 프로그램입니다. 메모장에서도 한글을 칠 수 있지만, 워드로 작성하면 더 편한 것과 같은 맥락입니다. 이런 친구들을 IDE(Integrated development environment, 통합개발환경)라고 합니다.

▼ 7) 파이썬 실행하기 & 첫 파이썬 파일 실행

- 파이썬 실행하기



Create New Project → 바탕화면의 **수업 폴더** 내 프로젝트명은 마음대로 적기



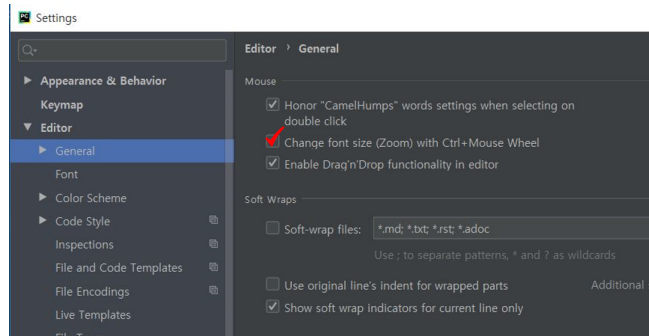
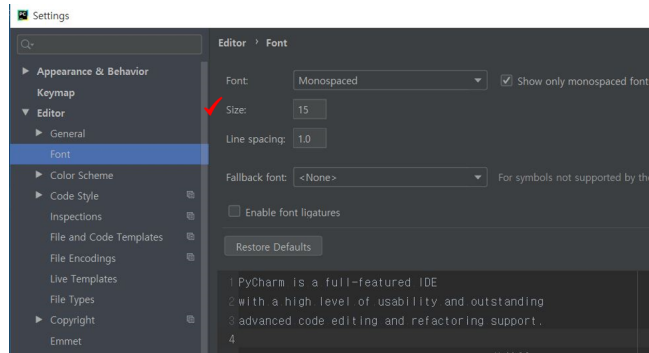
- 파이썬 파일을 하나 만들어 실행해보기



hello.py라는 파일에 아래와 같이 적어봅니다.

```
print ('hello world!')
```

▼ 참고) 파이썬 폰트 사이즈 조절



▼ 8) 파이썬 문법을 시작하기에 앞서..

- 파이썬은 매우 직관적인 언어이고, 할 수 있는 것도 많습니다. 그런데, 개발자들도 모든 문법을 기억하기란 쉽지 않습니다. 오늘 배우는 것 외에 필요한 것들은 구글링해서 찾아보면 됩니다!
- 예를 들면, 이런곳들: <https://velog.io/@ground4ekd/Python-basic#목적>
- 단, tuple과 set 자료형은 기초 단계에선 다루지 않겠습니다. (몰라도 괜찮고, 많은 개발자들도 모릅니다)

▼ 9) 파이썬 기초 문법

▼ 변수 & 기본연산

```
a = 3      # 3을 a에 넣는다
b = a      # a를 b에 넣는다
a = a + 1  # a+1을 다시 a에 넣는다

num1 = a*b # a*b의 값을 num1이라는 변수에 넣는다
num2 = 99 # 99의 값을 num2이라는 변수에 넣는다

# 변수의 이름은 마음대로 지을 수 있음!
# 진짜 "마음대로" 짓는 게 좋을까? var1, var2 이렇게?
```

▼ 자료형

- 숫자, 문자형

```
name = 'bob' # 변수에는 문자열이 들어갈 수도 있고,
num = 12 # 숫자가 들어갈 수도 있고,

is_number = True # True 또는 False -> "Boolean"형이 들어갈 수도 있습니다.

#####
# 그리고 List, Dictionary 도 들어갈 수도 있죠. 그게 뭔지는 아래에서!
```

- 리스트 형 (Javascript의 배열형과 동일)

```
a_list = []
a_list.append(1) # 리스트에 값을 넣는다
a_list.append([2,3]) # 리스트에 [2,3]이라는 리스트를 다시 넣는다

# a_list의 값은? [1,[2,3]]
# a_list[0]의 값은? 1
```

```
# a_list[1]의 값은? [2,3]
# a_list[1][0]의 값은? 2
```

- Dictionary 형 (Javascript의 dictionary형과 동일)

```
a_dict = {}
a_dict = {'name':'bob', 'age':21}
a_dict['height'] = 178

# a_dict의 값은? {'name':'bob', 'age':21, 'height':178}
# a_dict['name']의 값은? 'bob'
# a_dict['age']의 값은? 21
# a_dict['height']의 값은? 178
```

- Dictionary 형과 List형의 조합

```
people = [{'name':'bob', 'age':20}, {'name':'carry', 'age':38}]

# people[0]['name']의 값은? 'bob'
# people[1]['name']의 값은? 'carry'

person = {'name':'john', 'age':7}
people.append(person)

# people의 값은? [{'name':'bob', 'age':20}, {'name':'carry', 'age':38}, {'name':'john', 'age':7}]
# people[2]['name']의 값은? 'john'
```

▼ 함수

- 함수의 정의 - 이름은 마음대로 정할 수 있음!

```
# 수학기제에서
f(x) = 2*x+3
y = f(2)
y의 값은? 7

# 참고: 자바스크립트에서는
function f(x) {
  return 2*x+3
}

# 파이썬에서
def f(x):
  return 2*x+3

y = f(2)
y의 값은? 7
```

- 함수의 응용

```
def sum_all(a,b,c):
  return a+b+c

def mul(a,b):
  return a*b

result = sum_all(1,2,3) + mul(10,10)

# result라는 변수의 값은?

def minus(a,b):
  return a-b

result2 = minus(mul(10,10),sum_all(1,2,3))

# result2라는 변수의 값은?
```

▼ 조건문

```
def oddeven(num): # oddeven이라는 이름의 함수를 정의한다. num을 변수로 받는다.
    if num % 2 == 0: # num을 2로 나눈 나머지가 0이면
        return True # True (참)을 반환한다.
    else: # 아니면,
        return False # False (거짓)을 반환한다.

def checkbob(name):
    if name == 'bob': # name이 'bob'이면 True를, 아니면 False를 반환해라
        return True
    else:
        return False
```

▼ 반복문



파이썬에서의 반복문은, 리스트의 요소들을 하나씩 꺼내쓰는 형태입니다.

```
def allsum(mylist): # allsum이라는 이름의 함수를 정의하고, mylist를 변수로 받습니다.
    sum = 0 # sum이라는 변수에 0을 넣습니다.
    for i in mylist: # mylist의 원소들을 처음부터 돌면서, i에 넣어 아래 식을 수행합니다.
        sum = sum + i # sum에다 직전 sum에다 원소를 더한 값을 넣는 거죠.
    return sum # 최종 sum을 반환합니다.

sth_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(allsum(sth_list)) # 45

# 또는, 자동으로 리스트를 생성해주는 파이썬 자체함수 range(숫자)를 사용하면,
sth_list_2 = range(10) # [0,1,2,3,4,5,6,7,8,9] 와 같음
print(allsum(sth_list_2)) # 45
# 로도 간단하게 할 수 있습니다.
```

다른 예제를 살펴봅시다.

```
people = [{'name': 'bob', 'age': 20},
          {'name': 'carry', 'age': 38},
          {'name': 'john', 'age': 7}]

# 모든 사람의 이름과 나이를 출력해봅시다.
for person in people:
    print(person['name'], person['age'])

# 이번엔, 반복문과 조건문을 응용한 함수를 만들어봅시다.
# 이름을 받으면, age를 리턴해주는 함수
def get_age(myname):
    for person in people:
        if person['name'] == myname:
            return person['age']
    return '해당하는 이름이 없습니다'

print(get_age('bob'))
print(get_age('kay'))
```

▼ 함께 연습해보기

1. 메일주소가 맞는지 판단하기

```
a = 'spartacodingclub@gmail.com'

#채워야하는 함수
def check_mail(s):
    ## 여기에 코딩을 해주세요

#결과값
print(check_mail(a))

#아래와 같이 출력됩니다
True
```

▼ 답

```
def check_mail(s):
    return s.find('@') > -1
```

2. 메일주소 찾기

```
a = 'spartacodingclub@gmail.com'

#채워야하는 함수
def get_mail(s):
    ## 여기에 코딩을 해주세요

#결과값
print(get_mail(a))

#아래와 같이 출력됩니다
gmail
```

▼ 정답

```
def get_mail(s):
    return s.split('@')[1].split('.')[0]
```

3. 리스트 안의 개수 세기

```
#입력값
a = ['사과', '감', '감', '배', '포도', '포도', '딸기', '포도', '감', '수박', '딸기']

#채워야하는 함수
def count_list(a_list):
    ## 여기에 코딩을 해주세요

#결과값
print(count_list(a))

#아래와 같이 출력됩니다
{'사과': 1, '감': 3, '배': 1, '포도': 3, '딸기': 2, '수박': 1}
```

▼ 정답

```
def count_list(a_list):
    result = {}
    for element in a_list:
        if element in result:
            result[element] += 1
        else:
            result[element] = 1
    return result
```

▼ 10) 파이썬 패키지 설치해보기 + 가상환경에 대한 설명

▼ 파이썬에선, "가상환경"이란 개념이 존재합니다. 파이썬은 간단하게 import해서 쓸 수 있는 라이브러리들이 아주 많은데요. 그렇다보니, 프로젝트 별로 라이브러리들을 관리해야 할 필요가 생깁니다. 파이썬에선 이러한 라이브러리들을 '패키지'라고 부릅니다.



Project A가 라이브러리 a,b,c,d만 쓰고, Project B는 d,e,f만 쓴다면, 굳이 a,b,c,d,e,f를 다 import 할 필요는 없겠죠?

또는,

Project A는 라이브러리 a의 (version 1.1)을 쓰고, Project B는 라이브러리 a의 (version 1.9)를 쓴다면, 나눠서 설치해둘 필요가 있겠죠?

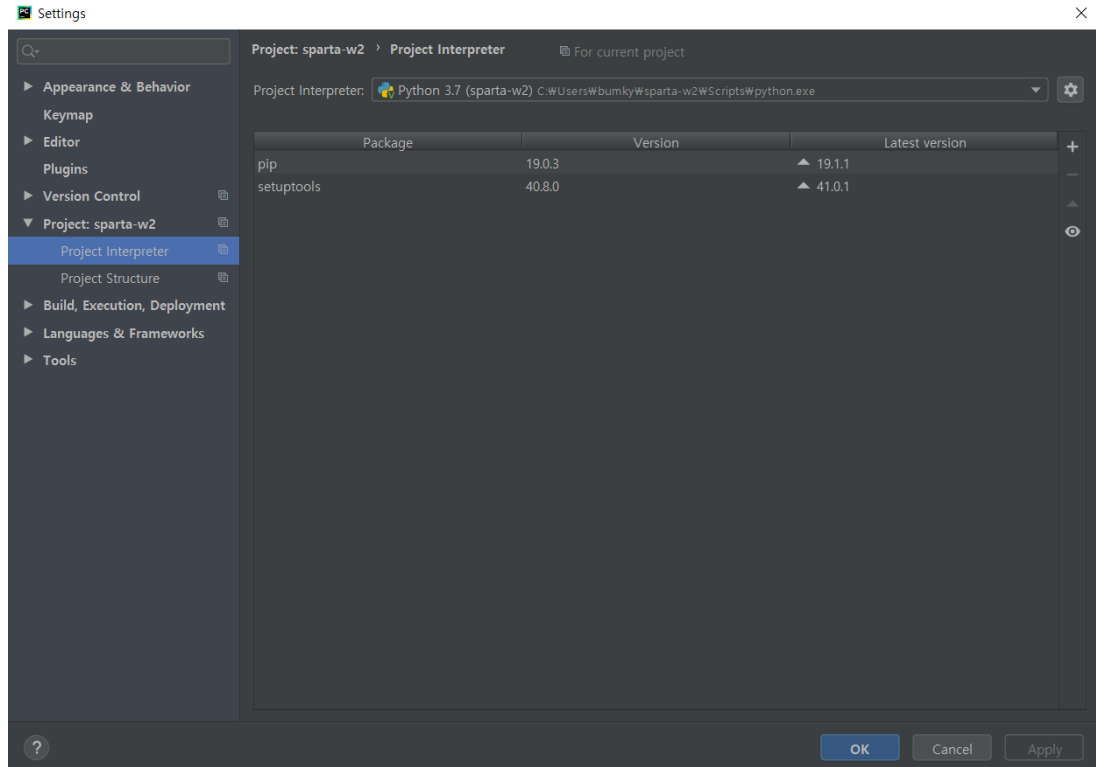


그러나 '가상환경'이라 이해하면 정말 virtual reality처럼 들어서 이해하기 어렵습니다. 사실 패키지를 부를 폴더를 선택해두는 것으로 이해하면 됩니다.

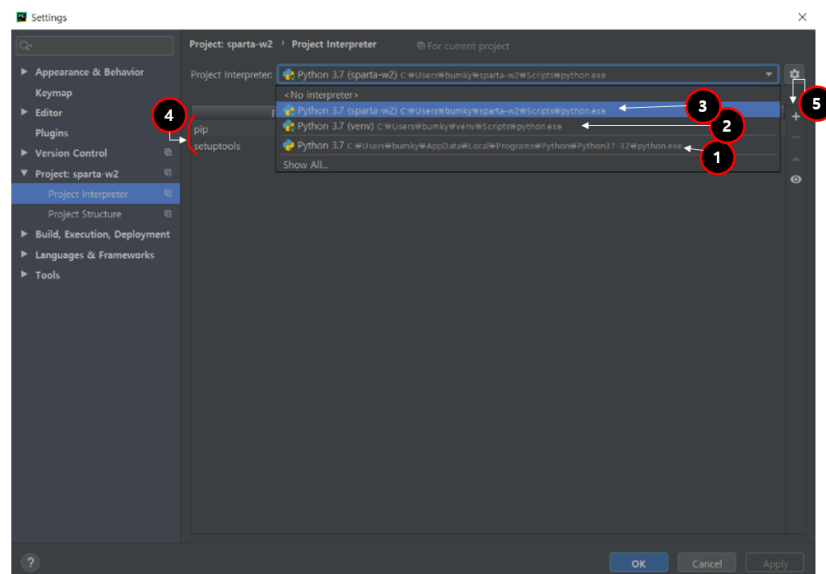
▼ 이렇게 설치합니다! (Request 패키지를 설치하고 싶은 경우)

▼ 윈도우: file → setting → project interpreter

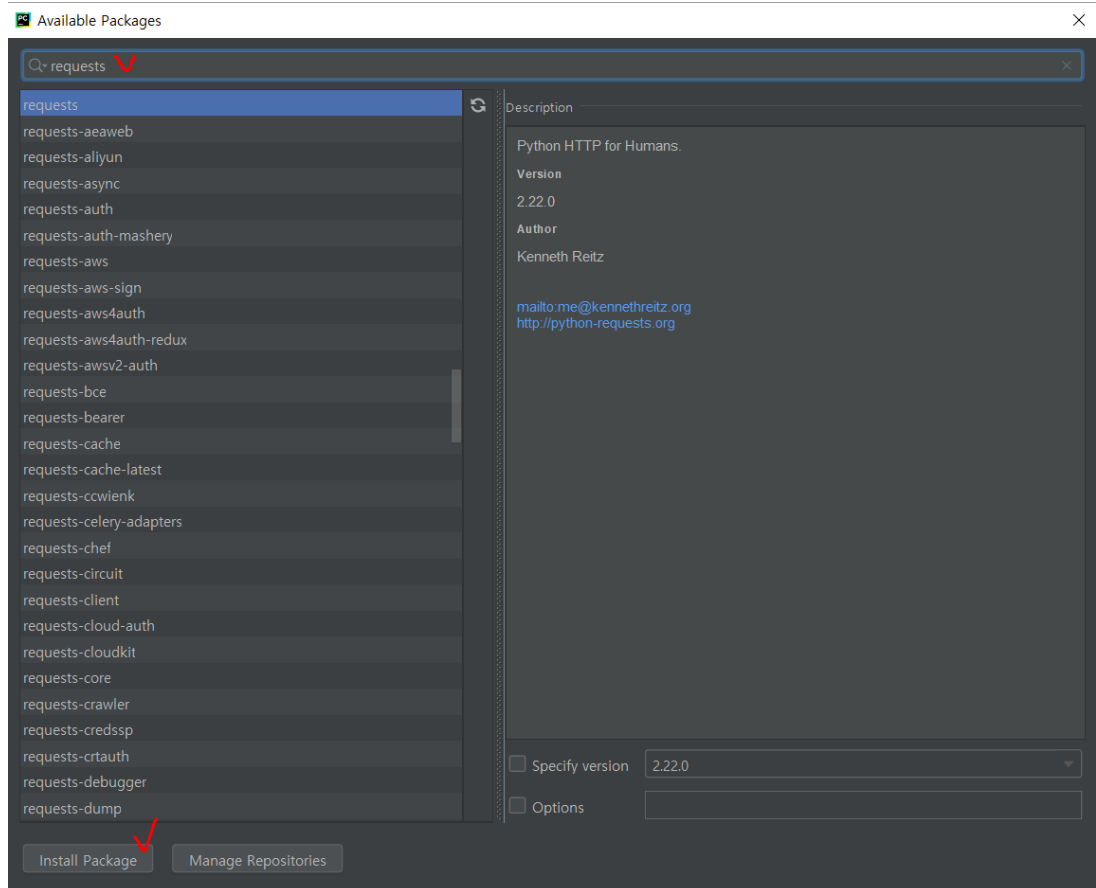
맥: preference → project interpreter로 접근



▼ 1: 가상환경이 아닌, 글로벌 파이썬을 의미합니다. / 2,3 : venv 그리고 sparta-w2라는 이름의 가상환경이 있네요! / 4: sparta-w2 가상환경에는 현재 pip와 setuptools라는 패키지만 설치되어 있습니다. / 5: 플러스 버튼을 눌러 다른 패키지를 설치할 수 있습니다.



▼ requests 패키지를 설치해볼게요! (s가 붙어있음에 주의!)



▼ 11) Requests 라이브러리 사용해보기 + List/Dictionary/함수/If/For문 연습

- 아래 방법으로 서울시 대기 OpenAPI에서, 종구의 NO2 값을 가져올 수 있음

```
import requests # requests 라이브러리 설치 필요

r = requests.get('http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99')
rjson = r.json()
print (rjson['RealtimeCityAir']['row'][0]['NO2'])
```

- 모든 구의 IDEX_MVL 값을 찍어주자!

```
import requests # requests 라이브러리 설치 필요

r = requests.get('http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99')
rjson = r.json()

gus = rjson['RealtimeCityAir']['row']

for gu in gus:
    print (gu['MSRSTE_NM'], gu['IDEX_MVL'])
```

- IDEX_MVL 값이 100 미만인 구만 찍어주자!

👉 들어쓰기가 얼마나 중요한지 다시 한번 확인해보세요!

```
import requests # requests 라이브러리 설치 필요

r = requests.get('http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99')
rjson = r.json()

gus = rjson['RealtimeCityAir']['row']
```



```
for gu in gus:
    if gu['IDEX_MVL'] < 100:
        print (gu['MSRSTE_NM'], gu['IDEX_MVL'])
```

[1시간]: 파이썬 갖고놀이1 - "웹 스크래핑"

▼ 12) 웹스크래핑 해보기 (영화 제목)

- 어떤 걸 스크래핑 할 계획인가요?



우리는 네이버영화의 아래 페이지를 스크래핑 하겠습니다.

<https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200303>

- 패키지 설치하기

```
beautifulsoup4, requests

- beautifulsoup4 -> HTML 파싱을 편리하게.
- requests -> 브라우저에서 엔터를 치는 효과. URL에 요청하기
```

- 크롤링 기본 세팅

```
import requests
from bs4 import BeautifulSoup

# 타겟 URL을 읽어서 HTML을 받아오고,
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200303', headers=headers)

# HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만들
# soup이라는 변수에 "파싱 용이해진 html"이 담긴 상태가 됨
# 이제 코딩을 통해 필요한 부분을 추출하면 된다.
soup = BeautifulSoup(data.text, 'html.parser')

#####
# (입맛에 맞게 코딩)
#####
```

- select의 사용법을 익혀봅니다.

```
import requests
from bs4 import BeautifulSoup

# URL을 읽어서 HTML을 받아오고,
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200303', headers=headers)

# HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만들
soup = BeautifulSoup(data.text, 'html.parser')

# select를 이용해서, tr들을 불러오기
movies = soup.select('#old_content > table > tbody > tr')

# movies (tr들) 의 반복문을 돌리기
for movie in movies:
    # movie 안에 a 가 있으면,
    a_tag = movie.select_one('td.title > div > a')
    if a_tag is not None:
        # a의 text를 찍어본다.
        print (a_tag.text)
```

- beautifulsoup 내 select에 미리 정의된 다른 방법을 알아봅니다

```
# 여러개를 가져오고 싶은 경우
soup.select('태그명')
soup.select('.클래스명')
soup.select('#아이디명')

soup.select('상위태그명 > 하위태그명 > 하위태그명')
soup.select('상위태그명.클래스명 > 하위태그명.클래스명')
```

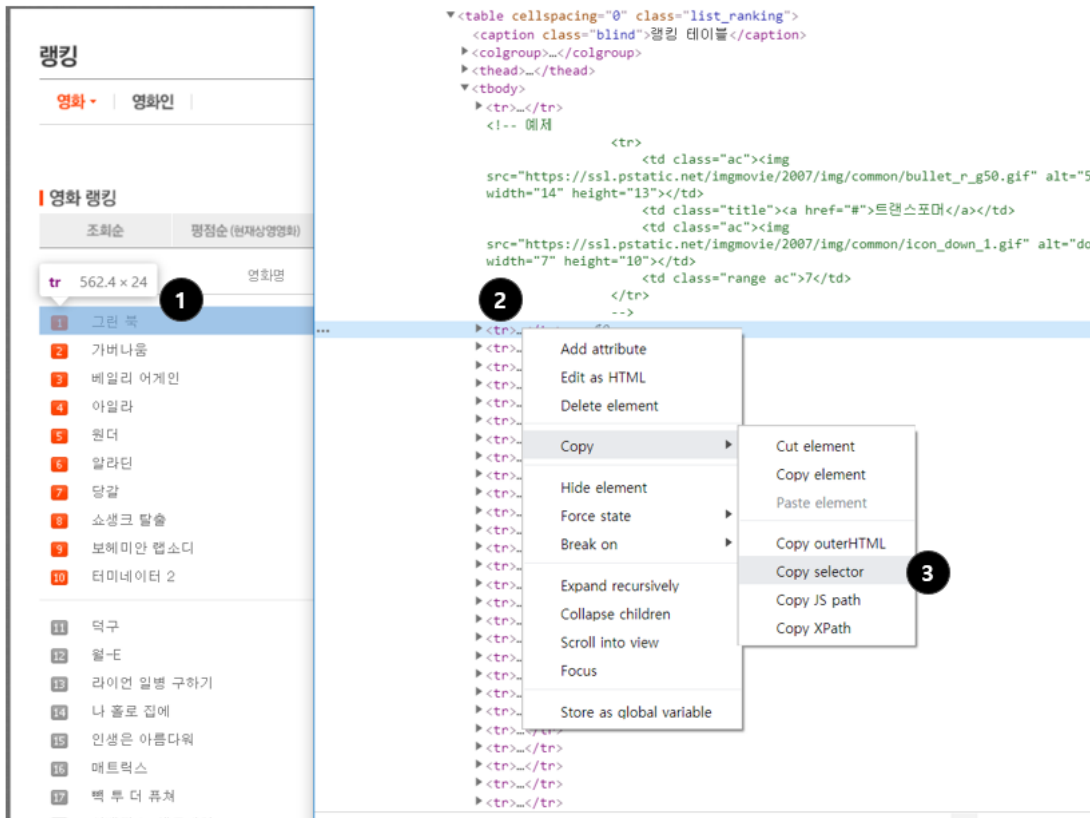
```
soup.select('태그명[속성="값"]')
```

```
# 한 개만 가져오고 싶은 경우
```

```
soup.select_one('위와 동일')
```

- 항상 정확하지는 않으나, 크롬 개발자도구를 참고할 수도 있습니다.

1. 원하는 부분에서 마우스 오른쪽 클릭 → 검사
2. 원하는 태그에서 마우스 오른쪽 클릭
3. Copy → Copy selector로 선택자를 복사할 수 있음



▼ 13) 웹스크래핑 추가 해보기 (순위, 영화 별점)

순위의 경우는, 굳이 스크래핑을 할 필요가 있을까요?
그냥 변수를 만들어 하나씩 증가시켜주면 됩니다!

▼ 예시: 아래와 같이 보이면 완성!

```
move_scrap x
bugger Console
Connected to pydev debugger (build 192.7142.56)
1 그린 북 9.60
2 가버나움 9.59
3 베일리 어게인 9.52
4 주전장 9.52
5 포트 V 페라리 9.51
6 아일라 9.49
7 원더 9.49
8 당갈 9.47
9 쇼생크 탈출 9.44
10 터미네이터 2:오리지널 9.43
```

▼ 코드

```
import requests
from bs4 import BeautifulSoup

# URL을 읽어서 HTML을 받아오고,
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200303', headers=headers)

# HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만들
soup = BeautifulSoup(data.text, 'html.parser')

# select를 이용해서, tr들을 불러오기
movies = soup.select('#old_content > table > tbody > tr')

# movies (tr들) 의 반복문을 돌리기
rank = 1
for movie in movies:
    # movie 안에 a 가 있으면,
    a_tag = movie.select_one('td.title > div > a')
    if a_tag is not None:
        title = a_tag.text
        star = movie.select_one('td.point').text
        print(rank, title, star)
        rank += 1
```

[1시간] : 소화 타임

▼ 일단 오늘 배운 것을 복습하고, 본격적으로 숙제를 시작합니다.

- 주요 키워드들: Ajax를 이용하는 전형적인 패턴을 이해하기, 파이썬 기초 문법(제일 중요한 건 "들여쓰기!"), beautifulsoup의 select 함수를 이용하는 것, for ... in ... 반복문 구문 헛갈리지 않기
- 키워드들을 바탕으로 튜터가 수업 내용을 간단히 복습해줍니다. (5분)

[끝]

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)
- 마찬가지로, 현재 본인의 감정상태와 수업후기에 관해 이야기합니다.
- 하단 숙제 & 설치해야 할 것들을 설명합니다.

숙제 & 설치

[숙제] - 다음 수업 D-1 까지 자신의 github에 올리고, url을 카톡방에 공유하기

- 1. 지니뮤직(<https://www.genie.co.kr/chart/top200?ditc=D&rtm=N&ynd=20200309>)의 1~50위 곡을 크롤링해보세요.

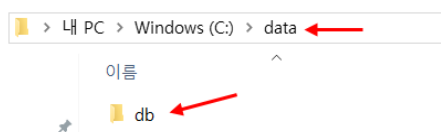
👉 순위 / 곡 제목 / 가수 (네이버영화 실습과 동일하게 진행)

[설치] - 다음 시간을 위해 미리 설치해야 할 것들

▼ mongoDB

▼ 윈도우

1. C드라이브에, 그림과 같이 data 라는 폴더를 만들고, 그 안에 db 라는 폴더를 만든다음, 설치 방법(→[설치 방법 링크](#))을 따라하세요.



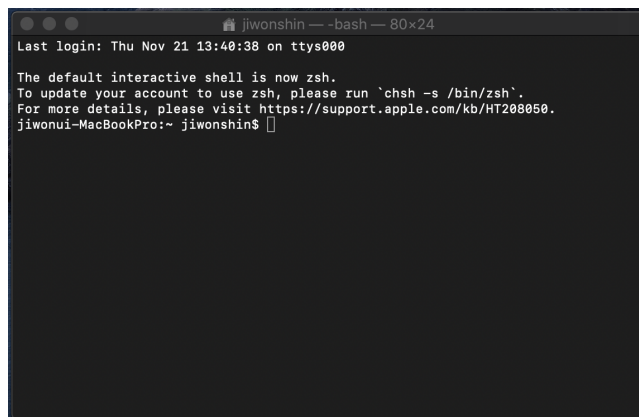
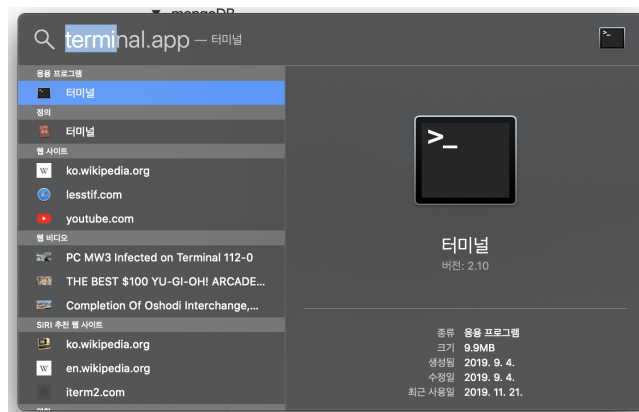
2. 그러면, 에러가 발생하지 않을 것이므로,
"아마 아래처럼 에러가 발생할것이다" → "이제 아래처럼 **wating for connections on port 27017** 라고 출력되면 정상적으로 실행된것이다. **27017** 는 포트번호이다."로 점프하시면 됩니다.

▼ 맥

1. 터미널을 연다. spotlight 에서, "terminal" 또는 "터미널"로 검색하면 됩니다!

👉 터미널이 무엇인가요?

우리는 주로 마우스를 써서 컴퓨터에 명령을 내리지만, 컴퓨터가 더 익숙한 건 '명령어'를 통한 명령입니다. 터미널은, '명령어를 쓸 수 있는 창'으로 생각해주시면 되겠습니다! 잘 여셨다면 아래와 같은 창을 만날 수 있습니다!



2. Homebrew 설치하기

👉 Homebrew는 무엇인가요?

새로운 것이 참 많죠? 지금부터 맥의 편리함에 빠져드신 겁니다.
Homebrew는 '다운로드 패키지'를 관리할 수 있는 툴이에요.

brew install 프로그램이름

을 입력하면, 프로그램을 자동으로 다운로드 받아 설치해준답니다. 엄청나죠?

터미널 창에 아래 코드를 복사, 붙여넣기 하고 엔터!

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

그러면 아래와 같은 화면이 나옵니다. 이 화면에서 다시 엔터!

```
jjiwonshin — ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only. T...
Last login: Thu Nov 21 13:40:38 on ttys000

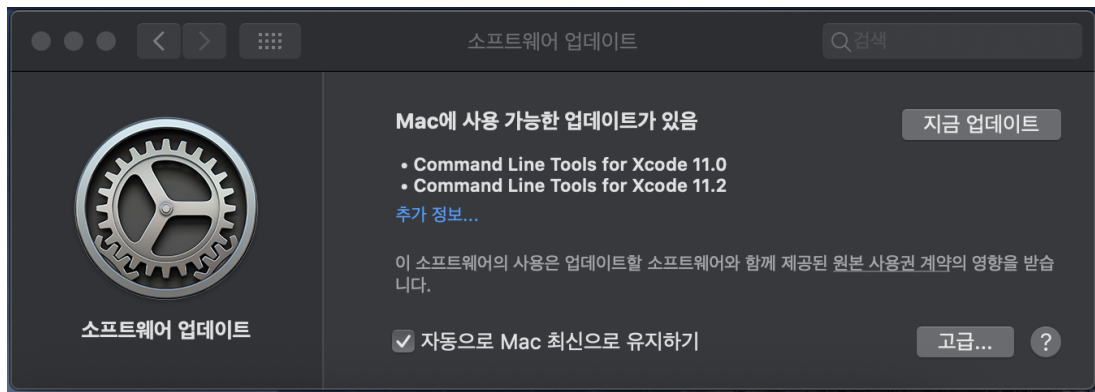
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[jiwonui-MacBookPro:~ jjiwonshin$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following new directories will be created:
/usr/local/bin
/usr/local/etc
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks
==> The Xcode Command Line Tools will be installed.

Press RETURN to continue or any other key to abort
█
```

그러면 아래와 같이 비밀번호 입력칸이 나옵니다. 내 맥북 비밀번호를 입력하고 엔터!
(비밀번호를 입력하더라도 화면에 나타나진 않아요. 사실 잘 입력되고 있으니 걱정 마세요.)

```
Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /usr/local/bin /usr/local/etc /usr/local/include
    /usr/local/lib /usr/local/sbin /usr/local/share /usr/local/var /usr/local/opt /
usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew
/usr/local/var/homebrew/linked /usr/local/Cellar /usr/local/Caskroom /usr/local/
Homebrew /usr/local/Frameworks
Password: █
```

엔터 누른 후에 만약 맥 소프트웨어(Xcode) 업데이트 알림이 뜨면 업데이트를 해주세요.



모든 설치가 완료되면 아래와 같은 화면이 나옵니다. Homebrew 설치 끝!

```

jiwonshin — -bash — 80x39
* [new tag]          2.1.12      -> 2.1.12
* [new tag]          2.1.13      -> 2.1.13
* [new tag]          2.1.14      -> 2.1.14
* [new tag]          2.1.15      -> 2.1.15
* [new tag]          2.1.16      -> 2.1.16
* [new tag]          2.1.2       -> 2.1.2
* [new tag]          2.1.3       -> 2.1.3
* [new tag]          2.1.4       -> 2.1.4
* [new tag]          2.1.5       -> 2.1.5
* [new tag]          2.1.6       -> 2.1.6
* [new tag]          2.1.7       -> 2.1.7
* [new tag]          2.1.8       -> 2.1.8
* [new tag]          2.1.9       -> 2.1.9
HEAD is now at fb978a786 Merge pull request #6759 from davidrupp/patch-1
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
    https://github.com/Homebrew/brew#donations
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Enumerating objects: 5087, done.
remote: Counting objects: 100% (5087/5087), done.
remote: Compressing objects: 100% (4880/4880), done.
remote: Total 5087 (delta 47), reused 319 (delta 10), pack-reused 0
Receiving objects: 100% (5087/5087), 4.15 MiB | 2.99 MiB/s, done.
Resolving deltas: 100% (47/47), done.
Tapped 2 commands and 4868 formulae (5,129 files, 12.8MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
    https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
    https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
    https://docs.brew.sh
jiwonui-MacBookPro:~ jiwonshin$

```

3. mongoDB 설치하기

터미널 창에 아래 코드 입력 후 엔터 (한줄씩 복사-붙여넣기 하세요)

```

brew tap mongodb/brew

```



```
brew install mongodb-community
```

몽고DB 설치가 완료되면 다음 화면이 나옵니다.

```
jwshin — -bash — 80x39
Receiving objects: 100% (5087/5087), 4.15 MiB | 2.99 MiB/s, done.
Resolving deltas: 100% (47/47), done.
Tapped 2 commands and 4868 formulae (5,129 files, 12.8MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh

jiwonui-MacBookPro:~ jiwonshin$ brew tap mongodb/brew
==> Tapping mongodb/brew
Cloning into '/usr/local/Homebrew/Library/Taps/mongodb/homebrew-brew'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 4), reused 5 (delta 1), pack-reused 0
Unpacking objects: 100% (13/13), done.
Tapped 6 formulae (45 files, 59.8KB).
jiwonui-MacBookPro:~ jiwonshin$
jiwonui-MacBookPro:~ jiwonshin$ brew install mongodb-community
==> Installing mongodb-community from mongodb/brew
==> Downloading https://fastdl.mongodb.org/osx/mongodb-macos-x86\_64-4.2.1.tgz
##### 100.0%
==> Caveats
To have launchd start mongodb/brew/mongodb-community now and restart at login:
  brew services start mongodb/brew/mongodb-community
Or, if you don't want/need a background service you can just run:
  mongod --config /usr/local/etc/mongod.conf
==> Summary
📦 /usr/local/Cellar/mongodb-community/4.2.1: 21 files, 273.5MB, built in 1 min
   ute 26 seconds
jiwonui-MacBookPro:~ jiwonshin$
```

4. mongoDB 실행해보기

```
brew services start mongodb-community
```

아래와 같은 화면이 나옵니다.

```
(base) Byeongkwanui-MacBook-Pro:~ byeongkwan$ brew services start mongodb-community
==> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-commu
```

! 앗, 혹시 brew services start mongodb-community 실행이 안되시나요?

▼ 안될 경우 대처 방안

아래 내용을 차례대로 실행해보세요

```
brew update
```

```
brew services start mongodb-community
```

5. 마지막으로, mongoDB 실행이 잘 되었는지 확인하기

☞ <http://localhost:27017> 에 접속했을 때,
아래와 같은 화면을 만나면 잘 된 것입니다!

← → ↻ ⓘ localhost:27017

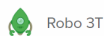
It looks like you are trying to access MongoDB over HTTP on the native driver port.

이제 mongoDB 설치가 모두 완료되었습니다.

▼ Robo3T

- 다운로드: <https://robomongo.org/download>

▼ Download Robo 3T 를 받으세요! (⇒ Studio 3T를 받지 마세요!)



Download

Blog

Account

The most powerful option

Need a more powerful GUI?
Meet Robo 3T's sibling, **Studio 3T**

- Enjoy rich query autocompletion
- Build queries via drag-and-drop
- Write SQL to query MongoDB
- Break down aggregation queries into stages
- Generate driver code in five languages
- Compare collections and view differences side-by-side
- Explore data schema and find outliers
- Import and export in multiple formats, including SQL Migration
- Automate MongoDB tasks like imports and data sync

Download Studio 3T

The latest version

Robomongo is now Robo 3T

Robo 3T 1.3 brings you support for MongoDB 4.0 and SCRAM-SHA-256, an upgraded mongo shell, support for importing from MongoDB SRV connection strings, among many other fixes and improvements.

Download Robo 3T