



[스파르타코딩클럽] 웹개발 종합반 - 3주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

1. 파이썬 기초 문법을 안다.
2. 원하는 페이지를 크롤링 할 수 있다.
3. pymongo를 통해 mongoDB를 제어할 수 있다.

[목차]

- 01. 오늘 배울 것
- 02. 연습 겸 복습 - 나홀로메모장에 OpenAPI 붙여보기
- 03. 파이썬 시작하기
- 04. 파이썬 기초공부
- 05. 파이썬 패키지 설치하기
- 06. 패키지 사용해보기
- 07. 웹스크래핑(크롤링) 기초
- 08. Quiz_웹스크래핑(크롤링) 연습
- 09. DB설치 확인
- 10. DB개괄
- 11. pymongo로 DB조작하기
- 12. 웹스크래핑 결과 저장하기
- 13. Quiz_웹스크래핑 결과 저장하기
- 14. 끝 & 숙제 설명

01. 오늘 배울 것

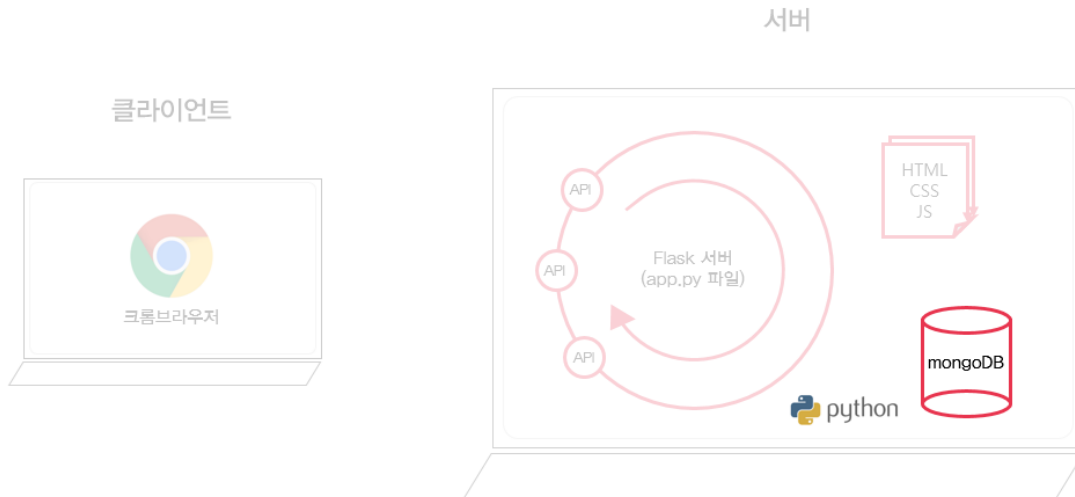
▼ 오늘 배울 것 이야기 - 3주차: Python, 크롤링, mongoDB



오늘은 드디어 '파이썬'을 배울거예요. 먼저 문법을 연습하고, 라이브러리를 활용하여 네이버 영화목록을 짹 가져와보겠습니다. (기대되죠!)

+

그리고, 우리의 인생 첫 데이터베이스. mongoDB를 다뤄볼게요!



02. 연습 겸 복습 - 나홀로메모장에 OpenAPI 붙여보기

🔥 2주차 리마인드를 위해 특별히 복습을 준비했어요. 한번 더 뽐내고 갑시다!

▼ 1) 스파르타가 만들어둔 OpenAPI 파악하기

▼ [코드스니펫] - 01. 나홀로메모장 API(GET)

```
http://spartacodingclub.shop/post
```

▼ [코드스니펫] - 02. 나홀로메모장 구경하기

```
http://spartacodingclub.shop/
```

☞ 네 그렇습니다!

나홀로메모장에 들어가는 **아티클들의 정보**를 불러오는 **OpenAPI**입니다.
이 API를 써서 저장된 포스팅 불러오기를 만들어볼게요!
(나중에 이 API를 실제로 만들어 볼 예정!)

▼ 2) 포스팅가져오기API 붙이기

1. 2주차에 완성했던 '나홀로메모장'을 켜봅시다!

▼ [코드스니펫] - 03. 코드가 없다면!

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

<head>
  <!-- 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">

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/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>

<title>스파르타코딩클럽 | 부트스트랩 연습하기</title>

<link href="https://fonts.googleapis.com/css2?family=Jua&display=swap" rel="stylesheet">

<style>
  * {
    font-family: 'Jua', sans-serif;
  }

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

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

  .posting-box {
    margin: 10px auto 30px auto;
    width: 500px;

    border: 3px solid black;
    border-radius: 5px;

    padding: 25px;

    display: none;
  }
</style>
<script>
  function openclose() {
    let status = $('#post-box').css('display');
    if (status == 'block') {
      $('#post-box').hide()
      $('#posting-box-btn').text('포스팅박스 열기')
    } else {
      $('#post-box').show()
      $('#posting-box-btn').text('포스팅박스 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">
        <a id="posting-box-btn" onclick="openclose()" class="btn btn-primary btn-lg" href="#"
          role="button">포스팅박스 열기</a>
      </p>
    </div>
    <div class="posting-box" id="post-box">
      <div class="form-group">
        <label for="exampleInputEmail1">아티클 URL</label>
        <input id="post-url" type="email" class="form-control" aria-describedby="emailHelp" placeholder="">
      </div>
      <div class="form-group">
        <label for="exampleInputPassword1">간단 코멘트</label>
        <input type="password" class="form-control" placeholder="">
      </div>
      <button type="submit" class="btn btn-primary">기사 저장</button>
    </div>
    <div class="card-columns" id="cards-box">
      <div class="card">
        
        <div class="card-body">
          <a href="http://naver.com" 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>
  </div>

```

```


<div class="card-body">
  <a href="http://naver.com" 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="http://naver.com" 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="http://naver.com" 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

▼ [코드스니펫] - 04. 로딩 후 바로실행

```

$(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++) {
      let article = articles[i];
      console.log(article);
    }
  }
})
```

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

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++) {
      let article = articles[i];
      make_card(article['comment'], article['desc'], article['image'], article['title'], article['url'])
    }
  }
})
```

8. 먼저 있던 카드들을 지워줍니다.

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

9. 1~8단계 통해 완성된 전체 script 코드

```
<script>
$(document).ready(function () {
```

```

        $('#cards-box').empty('');
        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++) {
                    let article = articles[i];
                    make_card(article['comment'], article['desc'], article['image'], article['title'], article['url'])
                }
            }
        })
    }

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

    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>

```

10. 완성된 코드 전체

▼ 코드

```

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

<head>
    <!-- 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">

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
        integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKUNimGFAV0wwj1yyf0RSJoZ+n"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
        integrity="sha384-wfSDF2E50Y2D1uUdj03uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
        crossorigin="anonymous"></script>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <title>스파르타코딩클럽 | 나홀로메모장</title>

    <!-- 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;
    }

    #post-box {
        width: 500px;
        margin: 20px auto;
        padding: 50px;
        border: black solid;
        border-radius: 5px;
        display: none;
    }
}
</style>
<script>
$(document).ready(function () {
    $('#cards-box').empty('');
    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++) {
                let article = articles[i];
                make_card(article['comment'], article['desc'], article['image'], article['title'], article['url']);
            }
        }
    })
}

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

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()" class="btn btn-primary btn-lg">포스팅박스 열기</button>
        </p>
    </div>
    <div class="form-post" id="post-box">
        <div>
            <div class="form-group">
                <label for="exampleFormControlInput1">아티클 URL</label>
                <input id="post-url" class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="exampleFormControlTextarea1">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
        </div>
    </div>
</div>

```

[illegible]

03. 파이썬 시작하기

- ### ▼ 3) 파이썬을 설치한다는 것의 의미

👉 파이썬을 설치한다?

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

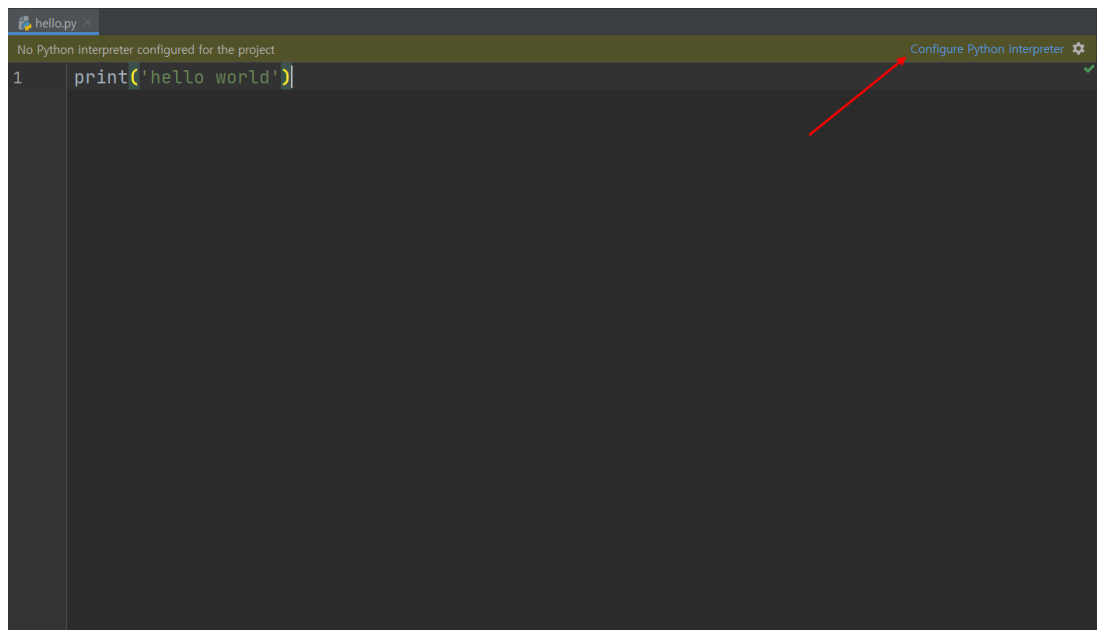
▼ 4) 첫 파이썬 파일 실행

- `hello.py` 파일 안에 다음 내용을 붙여넣습니다.

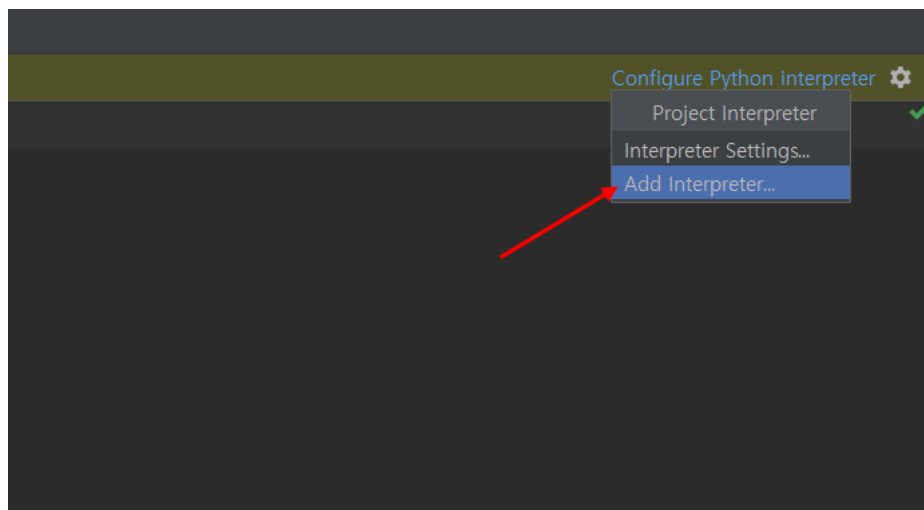
```
print ('Hello, sparta')
```

- 그럼 아래와 같은 메시지를 볼 수 있을 거예요! 파이썬 세팅을 해달라는 의미입니다.

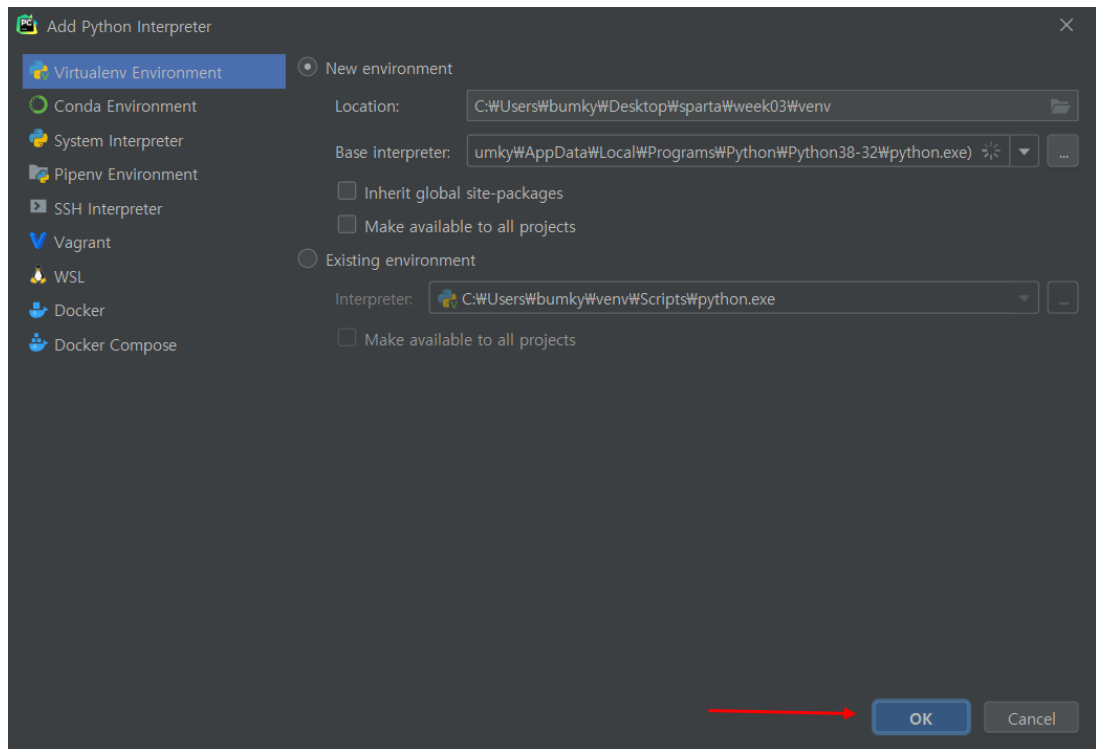
1. Configure Python Interpreter 클릭



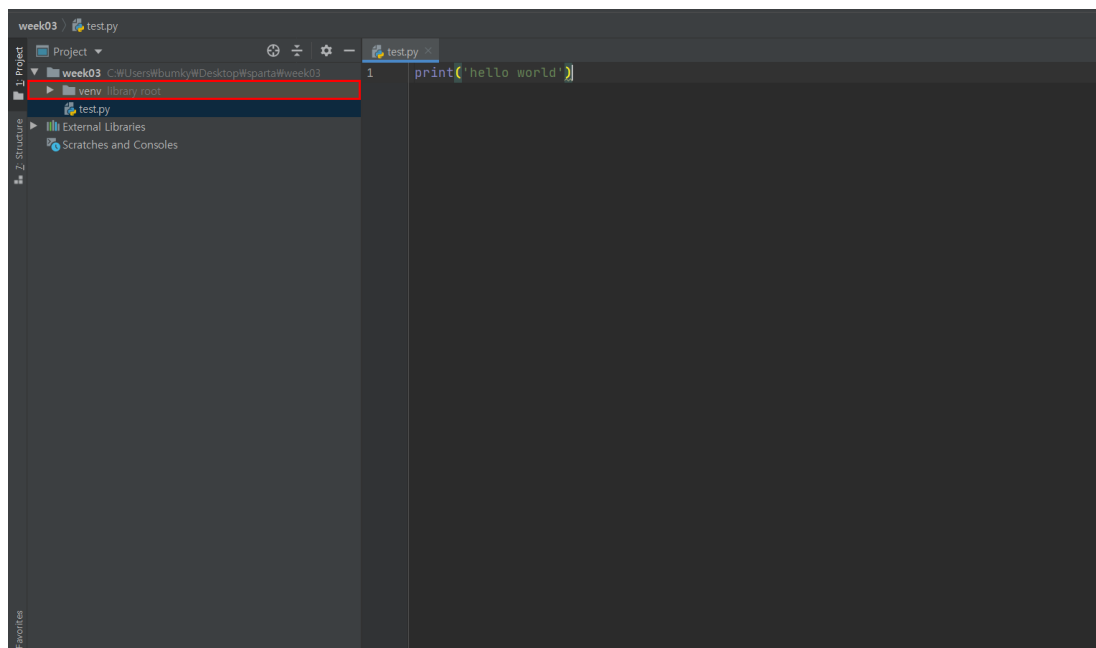
2. Add interpreter를 클릭!



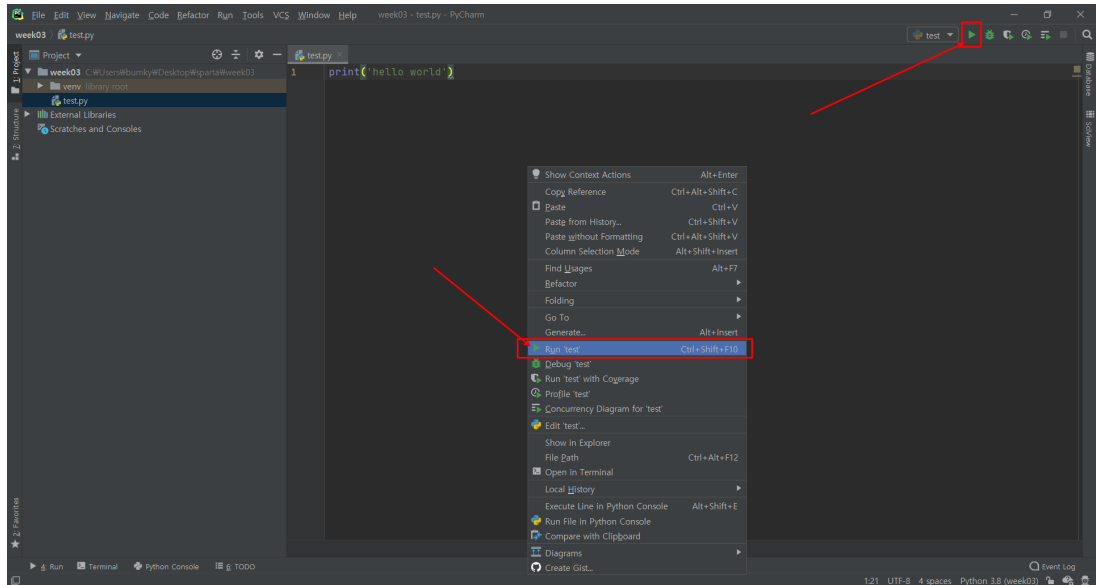
3. 그대로 OK를 클릭하면



4. 시간이 조금 흐르고(30~60초 정도) venv라는 폴더를 볼 수 있게 됩니다!



5. 마우스 오른쪽 버튼을 누르고 run을 클릭하면 실행! (또는 오른쪽 초록 삼각형 클릭!)



▼ 5) 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!
- 단, tuple과 set 자료형, class는 기초 단계에선 다루지 않겠습니다. (몰라도 괜찮고, 많은 개발자들도 모릅니다. 나중에 필요한 일이 생겼을 때 배워야 의미를 알 수 있습니다!)

04. 파이썬 기초공부

▼ 6) 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?
```

▼ 조건문

- if / else 로 구성!


```
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.

result = oddeven(20)
# result의 값은 무엇일까요?
```

```
def is_adult(age):
    if age > 20:
        print('성인입니다') # 조건이 참이면 성인입니다를 출력
    else:
        print('청소년이에요') # 조건이 거짓이면 청소년이에요를 출력

is_adult(30)
# 무엇이 출력될까요?
```

▼ 반복문

 파이썬에서의 반복문은, 리스트의 요소들을 하나씩 꺼내쓰는 형태입니다.

- 즉, 무조건 리스트와 함께 씁니다!

```
fruits = ['사과', '배', '감', '귤']

for fruit in fruits:
    print(fruit)

# 사과, 배, 감, 귤 하나씩 꺼내어 찍힙니다.
```

- 살짝 응용해볼까요? - 과일 갯수 세기 함수

▼ [코드스니펫] - 05. 리스트 예제

```
fruits = ['사과', '배', '배', '감', '수박', '귤', '딸기', '사과', '배', '수박']
```

```
fruits = ['사과', '배', '배', '감', '수박', '귤', '딸기', '사과', '배', '수박']

count = 0
for fruit in fruits:
    if fruit == '사과':
        count += 1

print(count)

# 사과의 갯수를 세어 보여줍니다.
```

```
def count_fruits(target):
    count = 0
    for fruit in fruits:
        if fruit == target:
            count += 1
    return count

subak_count = count_fruits('수박')
print(subak_count) #수박의 갯수

gam_count = count_fruits('감')
print(gam_count) #감의 갯수
```

- 다른 예제를 살펴봅시다.

▼ [코드스니펫] - 06. 딕셔너리 예제

```
people = [{'name': 'bob', 'age': 20},
          {'name': 'carry', 'age': 38},
          {'name': 'john', 'age': 7},
          {'name': 'smith', 'age': 17},
          {'name': 'ben', 'age': 27}]
```

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

```

        {'name': 'smith', 'age': 17},
        {'name': 'ben', 'age': 27}]

# 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'))

```

05. 파이썬 패키지 설치하기

▼ 7) 파이썬 패키지(package) 설치하기



패키지? 라이브러리? →

Python 에서 패키지는 모듈(일종의 기능들 묶음)을 모아 놓은 단위입니다. 이런 패키지 의 묶음을 라이브러리 라고 볼 수 있습니다. 지금 여기서는 외부 라이브러리를 사용하기 위해서 패키지를 설치합니다.

즉, 여기서는 **패키지 설치 = 외부 라이브러리 설치!**

▼ 1. 가상 환경(virtual environment) 이란? - 프로젝트별로 패키지들을 담을 공유함



문제상황:

회사에서는 패키지 A, B, C를 설치해서 쓰고,
개인 프로젝트에서는 패키지 B, C, D, E를 설치해서 쓰고 있었어요.

그런데 회사팀장님이 B를 이전 버전인 B' 로 쓰자고 하시네요.

그렇게 되면, 같은 컴퓨터에 깔려 있는 개인 프로젝트에서는 B' 로 쓰면 코드를 다 바꿔야 해요 😓

어떻게 하면 좋을까요?



해결책:

다 담아둘 필요 없이 공유함을 2개 만들어서,

공유함1에 A, B', C를 담아두고,
공유함2에 B, C, D, E를 담아두고 쓰면 관리하기 편하겠죠?

그래서, 가상환경이라는 개념이 등장했습니다.

즉, **프로젝트별 공유함** 이예요.



정리하자면,

가상환경(virtual environment)은

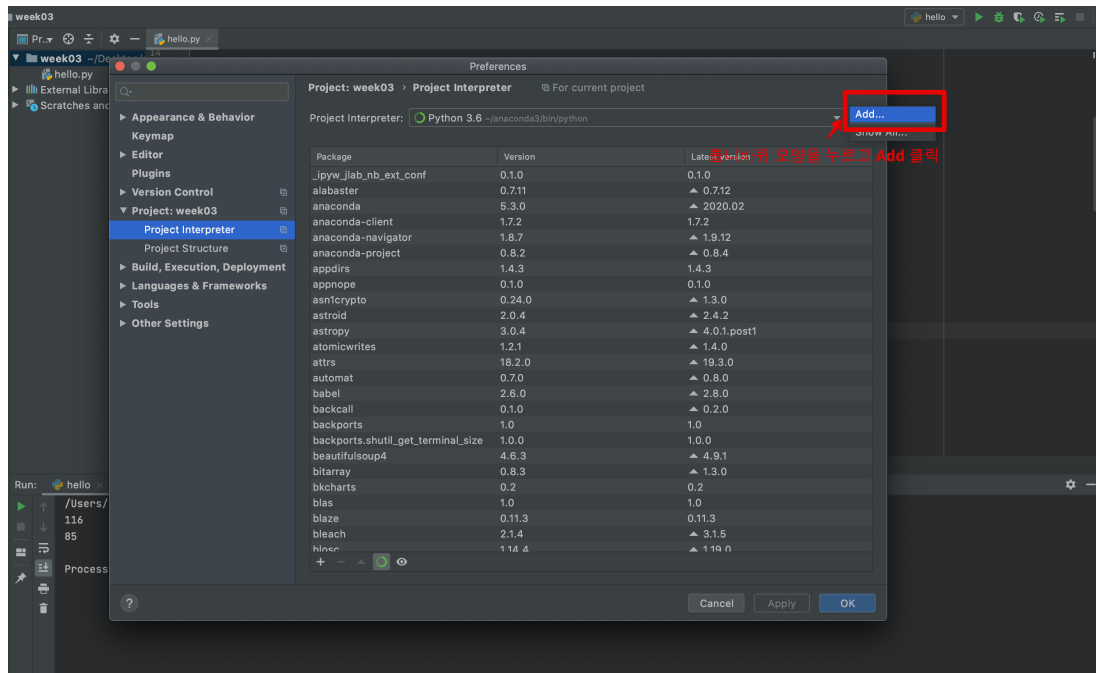
같은 시스템에서 실행되는 다른 파이썬 응용 프로그램들의 동작에 영향을 주지 않기 위해, 파이썬 배포 패키지들을 설치하거나 업그레이드하는 것을 가능하게 하는 **격리된 실행 환경** 입니다.

출처 : 파이썬 공식 용어집- 가상환경

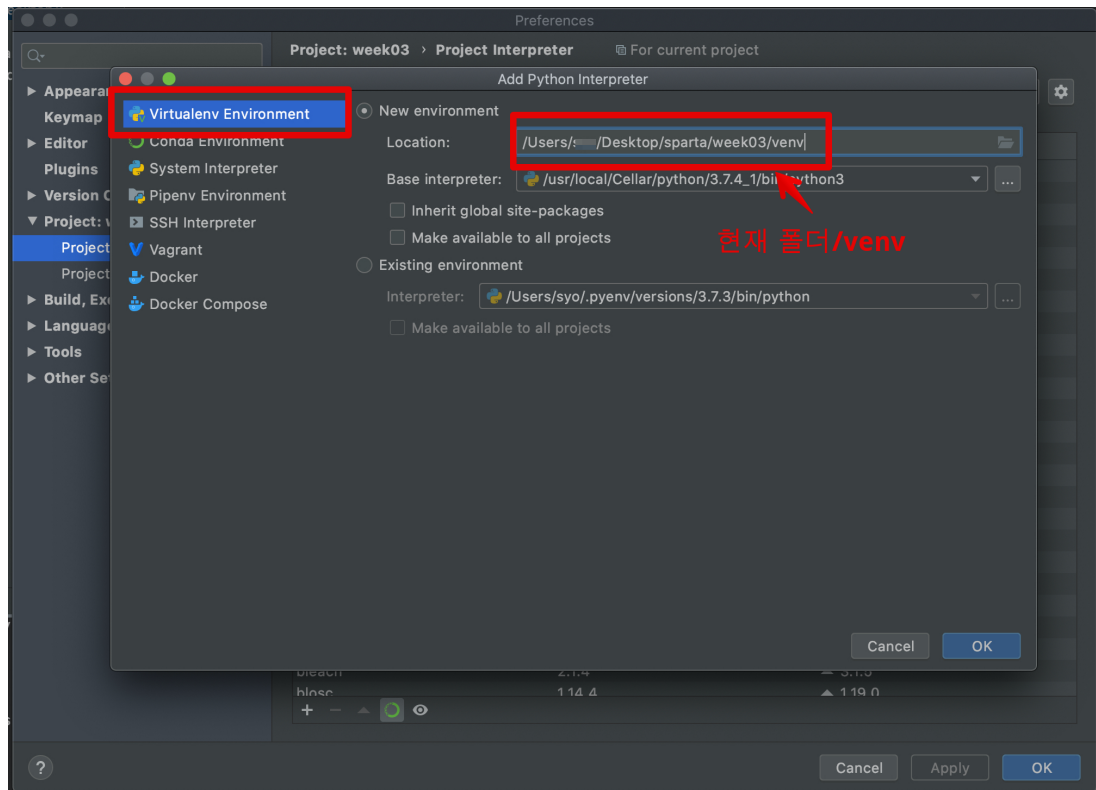
▼ 2. 가상 환경 설치 및 확인

👉 우린 이미 완료했으니, 확인만!

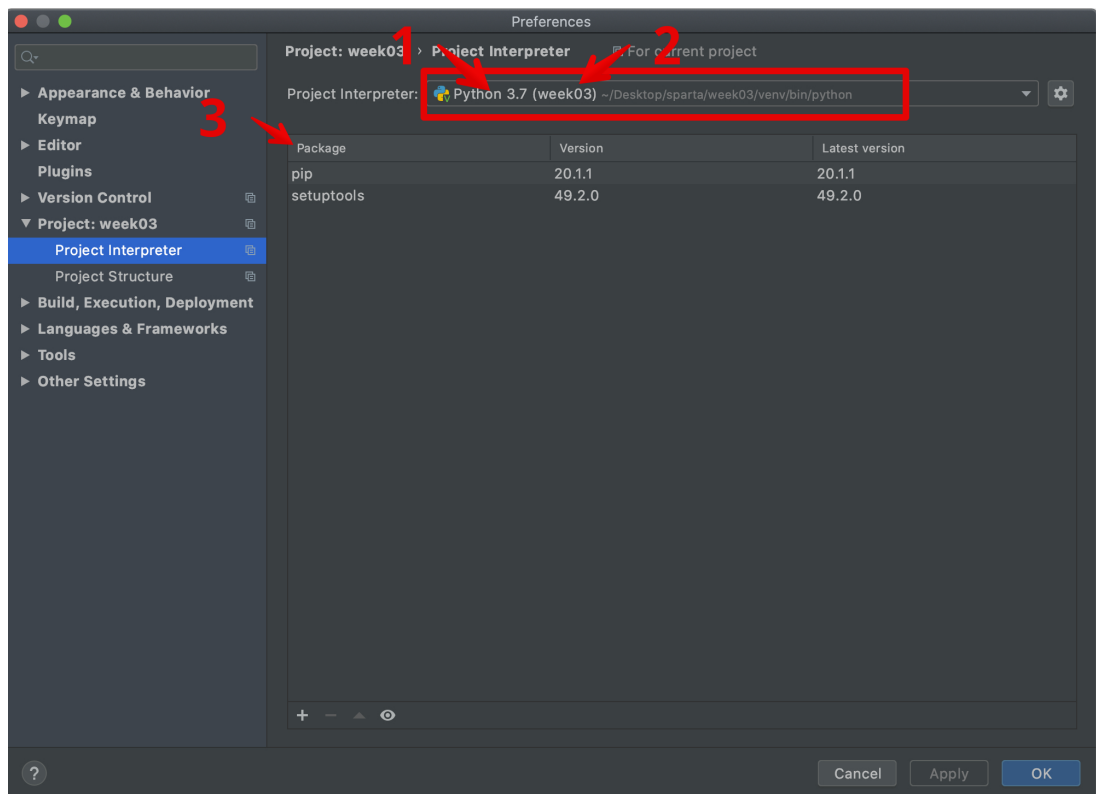
- Windows : File → settings → project interpreter
Mac : Preferences → Project Interpreter로 접근
 - 톱니바퀴 add 버튼 클릭



- virtual environment - New environment 선택
 - Location을 `현재폴더/venv` 로 설정해주세요. 현재 프로젝트 폴더(`week03`) 아래에 가상환경 폴더(`venv`)가 생성이 됩니다.



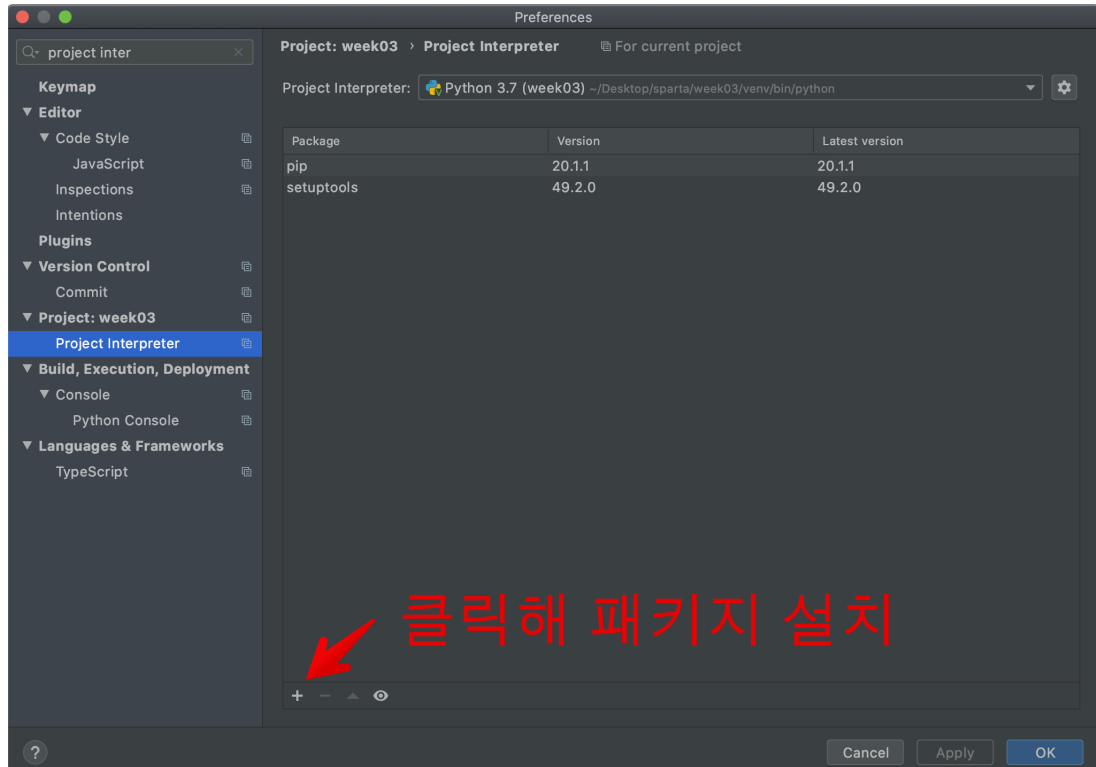
- 설정 부분을 해석하면
 1. 파이썬은 3.7 버전을 사용하고 있고
 2. **week03** 이라는 가상환경 명으로 현재폴더/venv 에 가상환경이 설정
 3. week03 가상환경에 설치된 패키지는 두 가지가 있네요.

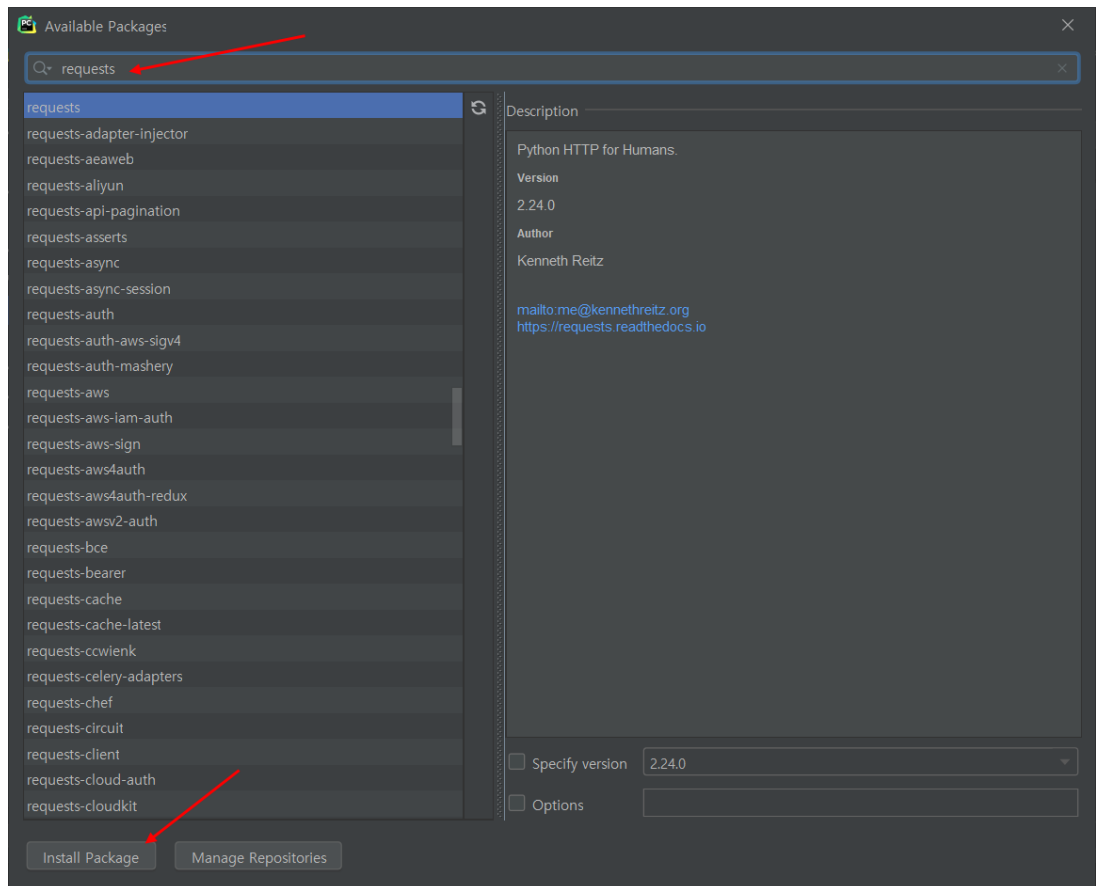


▼ 3. pip(python install package) 사용 - requests 패키지 설치해보기

👉 앱을 설치할 때 앱스토어/플레이스토어를 가듯이, 새로운 프로젝트의 라이브러리를 가상환경(공구함)에 설치하려면 pip 를 이용하게 됩니다.

- project interpreter 화면에서 + 버튼을 누르면 아래 창이 뜹니다!





06. 패키지 사용해보기

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

- 아래 방법으로 서울시 대기 OpenAPI에서, 중구의 NO2 값을 가져올 수 있습니다.

▼ [코드스니펫] - 07. requests 써보기

```
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 60 미만인 구만 찍어주자!

👉 들여쓰기가 얼마나 중요한지 다시한번 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'] < 60:
        print (gu['MSRSTE_NM'], gu['IDEX_MVL'])
```

07. 웹스크래핑(크롤링) 기초

▼ 9) 웹스크래핑 해보기 (영화 제목)

- 어떤 걸 스크래핑 할 계획인가요?

▼ [코드스니펫] - 08. 네이버영화페이지

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

- 패키지 추가 설치하기(BeautifulSoup4)

```
bs4
```

- 크롤링 기본 세팅

▼ [코드스니펫] - 09. 크롤링 기본 세팅

```
import requests
from bs4 import BeautifulSoup

headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200303', headers=headers)

soup = BeautifulSoup(data.text, 'html.parser')

# 코딩 시작
```

```
import requests
from bs4 import BeautifulSoup

# 타겟 URL을 읽어서 HTML를 받아오고,
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
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)
#####
```

- select / select_one의 사용법을 익혀봅니다.

👉 영화 제목을 가져와보기!

👉 태그 안의 텍스트를 찍고 싶을 땐 → 태그.text
태그 안의 속성을 찍고 싶을 땐 → 태그['속성']

```

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 만듦
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

```

# 선택자를 사용하는 방법 (copy selector)
soup.select('태그명')
soup.select('.클래스명')
soup.select('#아이디명')

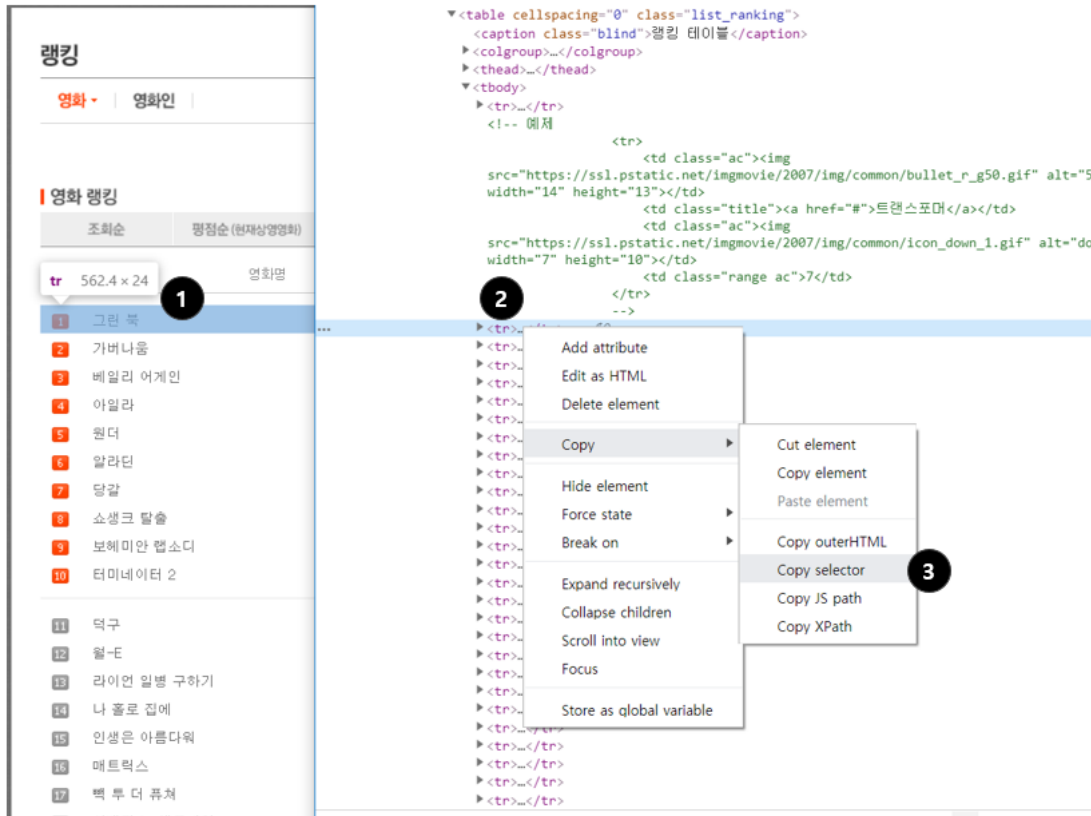
soup.select('상위태그명 > 하위태그명 > 하위태그명')
soup.select('상위태그명.클래스명 > 하위태그명.클래스명')

# 태그와 속성값으로 찾는 방법
soup.select('태그명[속성="값"]')

# 한 개만 가져오고 싶은 경우
soup.select_one('위와 동일')

```

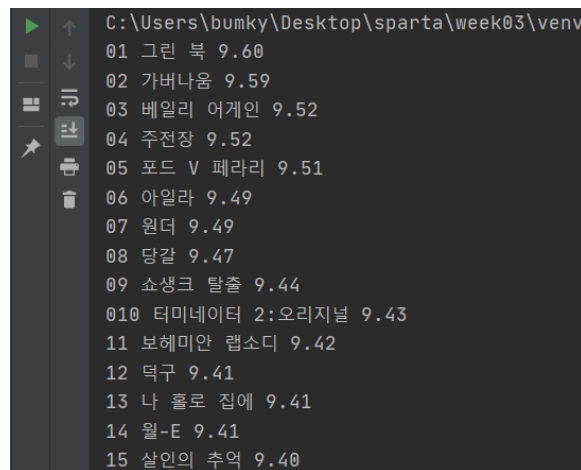
- 항상 정확하지는 않으나, 크롬 개발자도구를 참고할 수도 있습니다.
 1. 원하는 부분에서 마우스 오른쪽 클릭 → 검사
 2. 원하는 태그에서 마우스 오른쪽 클릭
 3. Copy → Copy selector로 선택자를 복사할 수 있음



08. Quiz_웹스크래핑(크롤링) 연습

▼ 10) 🐞 웹스크래핑 더 해보기 (순위, 제목, 별점)

▼ Q. 아래와 같이 보이면 완성!



▼ A. 완성 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
for movie in movies:
    # movie 안에 a 가 있으면,
    a_tag = movie.select_one('td.title > div > a')
    if a_tag is not None:
        rank = movie.select_one('td:nth-child(1) > img')['alt'] # img 태그의 alt 속성값을 가져오기
        title = a_tag.text # a 태그 사이의 텍스트를 가져오기
        star = movie.select_one('td.point').text # td 태그 사이의 텍스트를 가져오기
        print(rank, title, star)
```

09. DB설치 확인

▼ 11) DB 설치 확인

☞ 먼저, 각자 설치해온 DB가 잘 작동하는지 확인합니다. 크롬 창에 localhost:27017 이라고 쳤을 때, 아래와 같은 화면이 나오면 mongoDB가 돌아가고 있는 것입니다.



▼ 12) robo 3T 준비하기

• robo 3T의 역할

☞ mongoDB라는 프로그램은 참 특이한 친구예요. 눈으로 보이지 않는답니다. 유식한 말로, 그래픽인터페이스(=GUI)를 제공하지 않는다고 표현합니다.

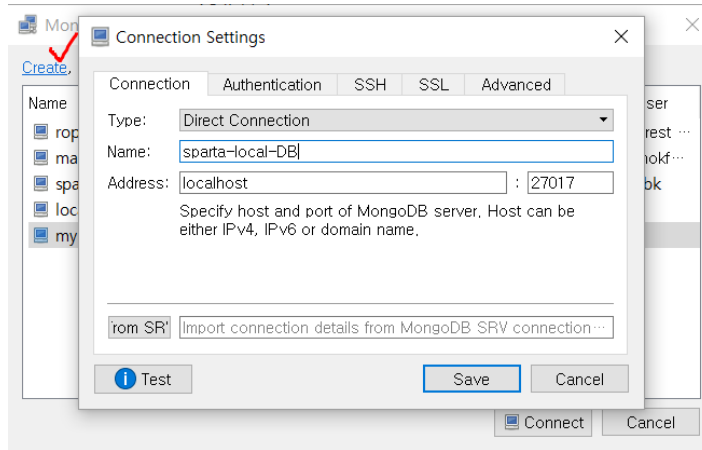
데이터를 저장했는데 눈으로 보이진 않고.. 답답하겠죠?

그래서 **DB내부를 살펴보기 위한 프로그램**을 따로 설치해야해요.
→ 이것이 바로 robo3T의 역할!

☞ 참고) 우리 눈에는 안보이지만(=GUI는 없지만) 컴퓨터에서 돌아가는 프로그램들은 무척 많으니, 너무 놀라지 마세요~!

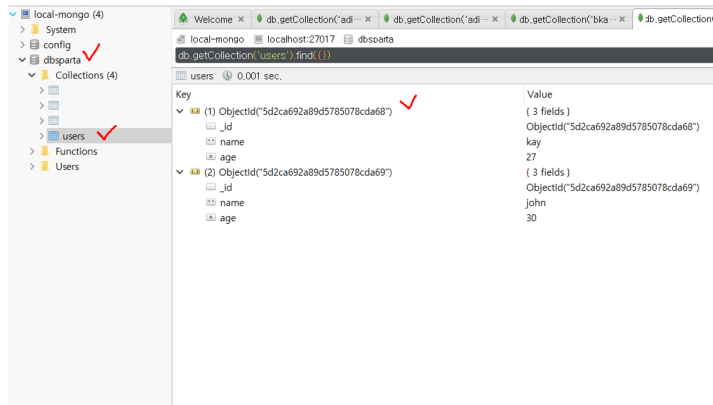
• robo3T 세팅하기

☞ 아래처럼 준비해서 robo3T에서 DB 볼 세팅을 미리 해둡니다!
(sparta-local-DB는 아무 이름이나 입력해도 됩니다)



- db, collection, documents(각 데이터들을 지칭)를 확인 가능합니다.

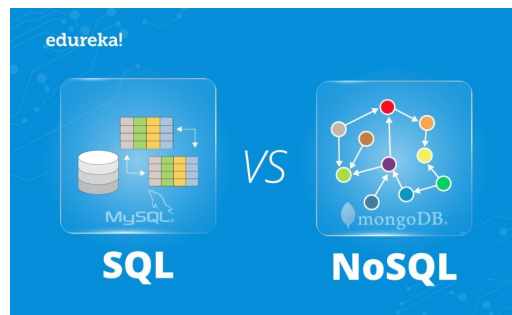
👉 지금은 System, config 외엔 아무것도 없죠?
조금 이따 데이터를 넣으면 아래처럼 보일거예요!



10. DB개괄

- ▼ 12) 들어가기 전에 : DB의 두 가지 종류

👉 Database에는, 크게 두 가지 종류가 있습니다.



☞ RDBMS(SQL)

행/열의 생김새가 정해진 엑셀에 데이터를 저장하는 것과 유사합니다. 데이터 50만 개가 적재된 상태에서, 갑자기 중간에 열을 하나 더하기는 어려울 것입니다. 그러나, 정형화되어 있는 만큼, 데이터의 일관성이나 / 분석에 용이할 수 있습니다.

ex) MS-SQL, My-SQL 등

☞ No-SQL

딕셔너리 형태로 데이터를 저장해두는 DB입니다. 고로 데이터 하나 하나 마다 같은 값들을 가질 필요가 없게 됩니다. 자유로운 형태의 데이터 적재에 유리한 대신, 일관성이 부족할 수 있습니다.

ex) MongoDB

11. pymongo로 DB조작하기

▼ 14) pymongo로 mongoDB 조작하기

- pymongo 라이브러리의 역할

☞ 예를 들어, MS Excel을 파이썬으로 조작하려면, 특별한 라이브러리가 필요하지 않겠어요?

마찬가지로, mongoDB 라는 프로그램을 조작하려면, 특별한 라이브러리, pymongo가 필요합니다!

- 패키지 설치하기

```
pymongo
```

▼ [코드스니펫] - 10. pymongo 기본 코드

```
from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbsparta

# 코딩 시작
```

- DB연결하기 & 데이터 넣기

```
from pymongo import MongoClient          # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta                    # 'dbsparta'라는 이름의 db를 만듭니다.

# MongoDB에 insert 하기

# 'users'라는 collection에 {'name':'bobby', 'age':21}를 넣습니다.
db.users.insert_one({'name':'bobby', 'age':21})
db.users.insert_one({'name':'kay', 'age':27})
db.users.insert_one({'name':'john', 'age':30})
```

- 모든 결과 값을 보기

```
from pymongo import MongoClient          # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta                    # 'dbsparta'라는 이름의 db를 만듭니다.
```



```
# MongoDB에서 데이터 모두 보기
all_users = list(db.users.find({}))

# 참고) MongoDB에서 특정 조건의 데이터 모두 보기
same_ages = list(db.users.find({'age':21}))

print(all_users[0])          # 0번째 결과값을 보기
print(all_users[0]['name'])  # 0번째 결과값의 'name'을 보기

for user in all_users:       # 반복문을 돌며 모든 결과값을 보기
    print(user)
```

- 특정 결과 값을 뽑아 보기

```
user = db.users.find_one({'name':'bobby'})
print(user)

# 그 중 특정 키 값을 빼고 보기
user = db.users.find_one({'name':'bobby'}, {'_id':False})
print(user)
```

- 수정하기

```
# 생김새
db.people.update_many( 찾을조건, { '$set': 어떻게바꿀지 })

# 예시 - 오타가 많으니 이 줄을 복사해서 씁시다!
db.users.update_one({'name':'bobby'}, {'$set':{'age':19}})

user = db.users.find_one({'name':'bobby'})
print(user)
```

- 삭제하기 (거의 안 씀)

```
db.users.delete_one({'name':'bobby'})

user = db.users.find_one({'name':'bobby'})
print(user)
```

▼ 15) pymongo 사용법. 코드요약

▼ [코드스니펫] - 11. pymongo 코드 요약

```
# 저장 - 예시
doc = {'name':'bobby', 'age':21}
db.users.insert_one(doc)

# 한 개 찾기 - 예시
user = db.users.find_one({'name':'bobby'})

# 여러개 찾기 - 예시 ( _id 값은 제외하고 출력)
same_ages = list(db.users.find({'age':21}, {'_id':False}))

# 바꾸기 - 예시
db.users.update_one({'name':'bobby'}, {'$set':{'age':19}})

# 지우기 - 예시
db.users.delete_one({'name':'bobby'})
```



우리는 딱 네 가지 기능만 알면 됩니다. 저장하고, 찾고, 바꾸고, 지우고!
이 기능들을 어떻게 사용하는지 요약하면 다음과 같습니다.

```
# 저장 - 예시
doc = {'name':'bobby', 'age':21}
db.users.insert_one(doc)

# 한 개 찾기 - 예시
user = db.users.find_one({'name':'bobby'})
```

```
# 여러개 찾기 - 예시 ( _id 값은 제외하고 출력)
same_ages = list(db.users.find({'age':21},{'_id':False}))

# 바꾸기 - 예시
db.users.update_one({'name':'bobby'},{'$set':{'age':19}})

# 지우기 - 예시
db.users.delete_one({'name':'bobby'})
```

12. 웹스크래핑 결과 저장하기

▼ 16) insert 연습하기 - 웹스크래핑 결과를 DB에 저장하기

- 이 코드에서 시작해봅시다!

▼ [코드스니펫] - 12. 크롤링 완성코드

```
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/73.0.3683.87 Safari/537.36'}
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:
        rank = movie.select_one('td:nth-child(1) > img')['alt'] # img 태그의 alt 속성값을 가져오기
        title = a_tag.text # a 태그 사이의 텍스트를 가져오기
        star = movie.select_one('td.point').text # td 태그 사이의 텍스트를 가져오기
        print(rank, title, star)
```

- pymongo 기본 세팅

```
import requests
from bs4 import BeautifulSoup

from pymongo import MongoClient # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta # 'dbsparta'라는 이름의 db를 만듭니다.

# URL을 읽어서 HTML을 받아오고,
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.87 Safari/537.36'}
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:
        rank = movie.select_one('td:nth-child(1) > img')['alt'] # img 태그의 alt 속성값을 가져오기
        title = a_tag.text # a 태그 사이의 텍스트를 가져오기
        star = movie.select_one('td.point').text # td 태그 사이의 텍스트를 가져오기
        print(rank, title, star)
```

- 도큐먼트 만들어 하나씩 insert 하기

```
import requests
from bs4 import BeautifulSoup
```

```

from pymongo import MongoClient          # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # MongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta                   # 'dbsparta'라는 이름의 db를 만듭니다.

# 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:
        rank = movie.select_one('td:nth-child(1) > img')['alt'] # img 태그의 alt 속성값을 가져오기
        title = a_tag.text # a 태그 사이의 텍스트를 가져오기
        star = movie.select_one('td.point').text # td 태그 사이의 텍스트를 가져오기
        doc = {
            'rank' : rank,
            'title' : title,
            'star' : star
        }
        db.movies.insert_one(doc)

```

13. Quiz_웹스크래핑 결과 저장하기

▼ 17) 🐞 find, update 연습하기 (delete는 연습 안할게요!)

- 파이썬 파일을 새로 하나 만들어 연습해봅니다.

▼ [코드스니펫] - 13. pymongo 기본 코드

```

from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbsparta

# 코딩 시작

```

- (1) 영화제목 '매트릭스'의 평점을 가져오기

▼ Q. 이렇게 되면 완성

```

C:\Users\bumky\Desktop
9.39
Process finished with...

```

▼ A. 완성 코드

```

from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbsparta

## 코딩 할 준비 ##

target_movie = db.movies.find_one({'title': '매트릭스'})
print(target_movie['star'])

```

- (2) '매트릭스'의 평점과 같은 평점의 영화 제목들을 가져오기

▼ Q. 이렇게 되면 완성

```
C:\Users\bumky\Desktop\sparta
인생은 아름다워
매트릭스
라이언 일병 구하기
사운드 오브 뮤직
헬프
포레스트 검프
안녕 베일리
글래디에이터
```

▼ A. 완성 코드

```
from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbsparta

## 코딩 할 준비 ##

target_movie = db.movies.find_one({'title': '매트릭스'})
target_star = target_movie['star']

movies = list(db.movies.find({'star': target_star}))

for movie in movies:
    print(movie['title'])
```

- (3) 매트릭스 영화의 평점을 0으로 만들기

▼ Q. 이렇게 되면 완성 (robo3T로 봤을 때)

13	ObjectId("5f...)	나 홀로 집에	13	9.41
14	ObjectId("5f...)	월-E	14	9.41
15	ObjectId("5f...)	살인의 추억	15	9.40
16	ObjectId("5f...)	백 투 더 퓨처	16	9.40
17	ObjectId("5f...)	인생은 아름다워	17	9.39
18	ObjectId("5f...)	매트릭스	18	0
19	ObjectId("5f...)	라이언 일병...	19	9.39
20	ObjectId("5f...)	사운드 오브...	20	9.39
21	ObjectId("5f...)	헬프	21	9.39
22	ObjectId("5f...)	포레스트 검프	22	9.39
23	ObjectId("5f...)	안녕 베일리	23	9.39
24	ObjectId("5f...)	글래디에이터	24	9.39
25	ObjectId("5f...)	위대한 쇼맨	25	9.38

▼ A. 완성 코드

```
from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbsparta

## 코딩 할 준비 ##

db.movies.update_one({'title': '매트릭스'}, {'$set': {'star': '0'}})
```

14. 끝 & 숙제 설명



지니뮤직의 1~50위 곡을 스크래핑 해보세요.

▼ [코드스니펫] - 14. 지니뮤직 사이트

<https://www.genie.co.kr/chart/top200?ditc=D&ymd=20200403&hh=23&rtm=N&pg=1>

- 순위 / 곡 제목 / 가수를 스크래핑 하면 됩니다.

▼ Q. 이렇게 되면 완성



```
Run: test x
1 아로하 조정석
2 시작 가호 (Gaho)
3 처음처럼 엠씨더맥스 (M.C the MAX)
4 이제 나만 믿어요 임영웅
5 아무노래 지코 (ZICO)
6 흔들리는 꽃들 속에서 네 샴푸향이 느껴진거야 장범준
7 원가 잘못됐어 권진아
8 WANNABE ITZY (있지)
9 돌덩이 하현우 (국카스텐)
10 어떻게 지내 (Prod. by VAN.C) 오반
11 METEOR 창모 (CHANGMO)
12 화려하지 않은 고백 규현 (KYUHYUN)
13 그때 그 아인 김필
14 Blueming 아이유 (IU)
15 문득 노을
16 마음을 드려요 아이유 (IU)
17 ON 방탄소년단
18 늦은 밤 너의 집 앞 골목길에서 노을
19 좋은 사람 있으면 소개시켜줘 조이 (JOY)
20 뚝 WINNER
21 Psycho Red Velvet (레드벨벳)
22 FIESTA IZ*ONE (아이즈원)
23 반만 진민호
```

👤 힌트:

순위와 곡제목이 깔끔하게 나오지 않을거예요. 옆에 여백이 있다던가, 다른 글씨도 나온다던가.. 파이썬 내장 함수인 `strip()`을 잘 연구해보세요!