



# 6~8주차 치트시트!



"이게 어디있었더라", "또 어떤 걸 할 수 있지?"

하는 분들을 위해 준비했습니다. 이름하여 치트시트!

## 주차 별 코드 요약

### HTML, CSS

#### ▼ 1) HTML 기본 골격



아무것도 표기되지 않은 가장 기본 형태입니다.

```
<html>
<head>
  <title>스파르타코딩클럽의 페이지</title>
  <!-- head 안에 들어 갈 어떤 요소들을 여기에 넣습니다 -->
</head>
<body>
  <!-- body 안에 들어 갈 어떤 요소들을 여기에 넣습니다 -->
</body>
</html>
```

#### ▼ 2) HTML class 선택자, 중첩



선택자를 살펴볼 수 있습니다.

1. mytitle이라는 class가 style에서 어떻게 정의되었고, body에서 어떻게 매칭되었는지 살펴봐주세요.
2. mytitle이라는 class와 small-font하는 class가 띄어쓰기로 중첩되었습니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>login!!!</title>
  <style>
    body {
      background-color: black;
      color: white;
    }
    .mytitle {
      color: yellow;
    }
    .small-font {
      font-size: 10px;
    }
  </style>
</head>
<body>
  <h1 class="mytitle small-font">
    This is login page.
  </h1>
  <div class="mytitle">
    ID:
    <input type="text">
    PW:
    <input type="text">
  </div>
  <button>로그인하기</button>
</body>
</html>
```

### ▼ 3) 자주쓰이는 CSS 모음



참고: <http://www.walterz.net/2017/08/18/자주-사용되는-css-속성/>

배경관련  
background-color  
background-image  
background-size

사이즈  
width  
height

폰트  
font-size  
font-weight  
font-family  
color

간격  
margin  
padding

### ▼ 4) Bootstrap 기본 골격



HTML 페이지를 시작할 때, 아래 코드를 복사해서 쓰세요!

```
<!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-G6eSO5c17kyS0qV26HWtO2D5D8+JG4K7Y74eRtQ3R2p6" crossorigin="anonymous">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKb+20U6N9s8x4Vceq+AJ0U+0IiLSc4+jFKEPl0nnQ1PJka9GCman5v1eBg/3+ulAZi8Q==" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6ulPEgpT9uV4jgMBZsp8f05ULjsg64Qb1reg5Maq616MC36ws30A"></script>

  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

### ▼ 5) Bootstrap 가져다쓰기



참고: <https://getbootstrap.com/docs/4.0/components/alerts/>  
components의 적절한 코드를 <body></body>에 넣으면 끝!

```
<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 class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
  </p>
</div>
```

## ▼ 6) 1주차 완성 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-Gn5"

  <!-- 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+Y1QKtv"
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vL

  <!-- 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;
    }
  </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 class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
      </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">
      <div class="card">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
      <div class="card">
        
  <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>
  <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
</div>
</div>
<div class="card">
  
    <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>
    <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
  </div>
</div>
<div class="card">
  
    <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>
    <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
  </div>
</div>
<div class="card">
  
    <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>
    <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
  </div>
</div>
</div>
</div>
</body>
</html>

```

## Javascript, JQuery, Ajax

▼ 1) Javascript 기초문법 (변수, 자료형, 함수, 반복문, 조건문, + 결합)

- 변수 선언

```
let first_name = 'bob' // 문자열
let age = 24 // 숫자
let is_even = true // 참거짓
```

- 자료형

```
let fruits = ['감', '배', '귤']
console.log(fruits[0]) // 감' 출력
fruits.push('수박') // 요소 추가
console.log(fruits[3]) // '수박' 출력
```

덕셔너리

```
let person = {'name':'bob','age':24}
person['height'] = 178 // 'height'라는 키 값에 178이라는 값 추가

console.log(person['name']) // 'bob' 출력
```

- 함수

'두 수를 더합니다' 라는 alert이 먼저 뜨고,  
num3 에는 7이 들어감

```
function sum(num1,num2){
```

```

    alert('두 수를 더합니다')
    return num1+num2
}

let num3 = sum(3,4)

```

- 반복문

```

let names = ['Bob', 'Kay', 'Juice']
for (let i = 0; i < names.length; i++) {
    console.log(names[i]); // i가 0부터 2까지 돌
}

```

- 조건문

```

if, else if, else

function compare_num(num){
    if (num > 90) {
        alert('90보다 커요!')
    } else if (num > 80) {
        alert('80보다 커요!')
    } else {
        alert('80보다 작아요!')
    }
}

```

**AND** 조건은 "&&"로,  
**OR** 조건은 "||"로 표현

**예)**

```

// 나이가 20보다 크고 30보다 작은 경우 true
function check_adult(age) {
    if (age > 20 && age < 30){
        return true;
    } else {
        return false;
    }
}

// 나이가 20보다 작거나, 30보다 크면 true
function check_adult(age) {
    if (age < 20 || age > 30){
        return true;
    } else {
        return false;
    }
}

```

- 결합

**만약에, 0부터 n-1까지 더하는 함수를 만들고 싶다면?**

```

function get_sum(n) {
    let result = 0
    for (let i = 0; i < n; i++) {
        result = result + i;
        // 반복문을 돌면서, result에다 i를 더해줍니다.
        // result += i; 라고 줄여 쓸 수 있습니다.
    }
    return result
}

console.log(get_sum(10)) // 45를 출력

```

**리스트의 값을 모두 출력하게 하고 싶으면?**

```

let names = ['Bob', 'Kay', 'Juice']
for (let j = 0; j < names.length; j++) { // names 리스트의 길이 미만까지 돌아라!
    console.log(names[j]);
}

```

반복문과 조건문의 결합  
리스트의 값을 돌면서 나이가 20세 미만인 사람을 출력하기

```
let people = [{ 'name': 'bob', 'age': 24 },
  { 'name': 'james', 'age': 18 },
  { 'name': 'kay', 'age': 30 },
  { 'name': 'john', 'age': 35 } ]

for (let i = 0 ; i < people.length; i++) {
  if (people[i]['age'] < 20) {
    console.log(people[i]);
  }
}
```

## ▼ 2) HTML 버튼과 Javascript 연결해보기

👉 버튼의 onclick 함수에 자바스크립트의 hey() 함수를 부르고 있습니다.

1, 2, 3, 4 .. 높여가며 숫자를 보여주는 onclick 함수 만들어보기

```
//자바스크립트 부분
<script>
  let num = 0;

  function hey(){
    num += 1;
    if (num > 20) {
      alert("안녕! " + num + "번 누르셨네요!");
    } else {
      alert('앗. 그만 누르세요 이제! ');
    }
  }
</script>

//버튼 부분
<button onclick="hey()" class="btn btn-primary">기사저장</button>
```

## ▼ 3) HTML 로딩 후 Javascript 실행하기

```
<script>

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

function listing() {
  alert('이 함수는 HTML 로딩 후 불러옵니다.')
}

</script>
```

## ▼ 4) JQuery 임포트하기

👉 참고: [https://www.w3schools.com/jquery/jquery\\_get\\_started.asp](https://www.w3schools.com/jquery/jquery_get_started.asp)  
<head> 와 </head> 사이에 아래를 넣으면 됩니다.  
(Bootstrap 기본 템플릿에 이미 다음과 같이 임포트 해두었습니다)

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

## ▼ 5) 자주 사용하는 JQuery 모음

### ▼ 1. input 박스의 값을 가져와보기

```
// 조작하고 싶은 부분에 id 값을 주기
<div class="form-post">
  <form>
    <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>
        <input class="form-control" placeholder="">
      </div>
      <button type="button" class="btn btn-primary">기사저장</button>
    </form>
  </div>

```

```

// 크롬 개발자도구 콘솔창에서 쳐보기
// id 값이 posting-url인 곳을 가리키고, val()로 값을 가져온다.
$('#posting-url').val();

```

## ▼ 2. div 보이기 / 숨기기

```

// 조작하고 싶은 부분에 id 값을 주기
<div id="posting-box" class="form-post">
  <form>
    <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>
      <input id="posting-comment" class="form-control" placeholder="">
    </div>
    <button onclick="posting()" type="button" class="btn btn-primary">기사저장</button>
  </form>
</div>

```

```

// 크롬 개발자도구 콘솔창에 쳐보기
// id 값이 posting-box인 곳을 가리키고, hide()로 안보이게 한다.(=css의 display 값을 none으로 바꾼다)
$('#posting-box').hide();

// show()로 보이게 한다.(=css의 display 값을 block으로 바꾼다)
$('#posting-box').show();

```

## ▼ 3. css의 값 가져와보기 (여기선 display 속성 값을 가져와볼게요!)

```

$('#posting-box').hide();
$('#posting-box').css('display');

$('#posting-box').show();
$('#posting-box').css('display');

```

## ▼ 4. 태그 내 텍스트 입력하기

### 1) input box의 경우

```

$('#posting-url').val('여기에 텍스트를 입력하면!');

```

### 2) 다른 것들 - 예) 버튼의 텍스트 바꾸기

```

// 가리키고 싶은 버튼에 id 값을 준다음
<button id="btn-posting-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>

```

```

$('#btn-posting-box').text('포스팅 박스 닫기');

```

## ▼ 5. 태그 내 html 입력하기

- <div> ~ </div> 내에,  
동적으로 html을 넣고 싶을 땐? (예를 들어, 포스팅되면 → 카드 추가)

```
// 사이에 html을 넣고 싶은 태그에 id값을 준 다음
<div id="cards-box" class="card-columns card-post">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">예를 들면 이렇게 카드가 나옵니다!!</h5>
      <p class="card-text">여기에 기사 내용이 들어가겠쥬</p>
      <p class="card-text comment">여기엔 코멘트가 들어갑니다</p>
    </div>
  </div>
</div>
```

#### 1) 버튼을 넣어보기

```
let temp_html = '<button>나는 추가될 버튼이다!</button>';
$('#cards-box').append(temp_html);
```

#### 2) 버튼 말고, 카드를 넣어보기

```
// 버튼과 같은 것인데, 좀 더 긴 html임. 줄 바꿈엔 \를 넣어줘야 이어질 수 있음
let temp_html = '<div class="card">\
  \
  <div class="card-body">\
    <h5 class="card-title">예를 들면 이렇게 카드가 나옵니다!!</h5>\
    <p class="card-text">여기에 기사 내용이 들어가겠쥬</p>\
    <p class="card-text comment">여기엔 코멘트가 들어갑니다</p>\
  </div>\
</div>';
$('#cards-box').append(temp_html);
```

### ▼ 6) Ajax로 API 호출하기

#### 👉 데이터를 담은 POST 요청

```
let url = 'https://www.nytimes.com/interactive/2016/08/25/universal/ko/36-hours-seoul-korean.html';
let comment = '코멘트 테스트';
let author = '스파르타코딩클럽';

$.ajax({
  type: "POST",
  url: "/post",
  data: { url_give:url, comment_give:comment, author_give:author },
  success: function(response){
    console.log(response)
  }
})
```

#### 👉 데이터를 담은 GET 요청

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

### ▼ 7) 2주차 완성 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-Gn5"

<!-- 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+Y1QKtv"
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vL

<!-- 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')
  }
}
function posting() {
  // 읽기
  let url = $('#posting-url').val();
  let comment = $('#posting-comment').val();
  let author = '스파르타코딩클럽';

  // 우리가 크롬 콘솔창에서 썼던 그 코드!
  $.ajax({
    type: "POST", // POST 방식으로 요청하겠다.
    url: "http://spartacodingclub.shop/post", // post라는 url에 요청하겠다.
    data: { url_give: url, comment_give: comment, author_give:author}, // 데이터를 주는 방법
    success: function(response){ // 성공하면
      if (response['result'] == 'success') {
        alert('포스팅 성공!');
        window.location.reload();
      } else {
        alert('서버 오류!')
      }
    }
  })
}
$(document).ready(function(){
  $('#cards-box').html('');
  listing();
});

function listing() {
  $.ajax({
    type: "GET",
    url: "http://spartacodingclub.shop/post?author_give=스파르타코딩클럽",
    data: {},
    success: function(response){
      if (response['result'] == 'success') {
        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]['
        }
      } else {
        alert('기사를 받아오지 못했습니다');
      }
    }
  })
}

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">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
      <div class="card">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
      <div class="card">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
      <div class="card">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
      <div class="card">
        
          <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>
          <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
        </div>
      </div>
    </div>
  </div>

```

```

        <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>
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
        <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>
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
</div>
</div>
</div>
</body>
</html>

```

## Python, MongoDB, 크롤링

### ▼ 1) 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!

```
def f(x):
    return 2*x+3
```

- 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? 106

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

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


# result2라는 변수의 값은? 94
```

## ▼ 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.

```
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'))
```

## ▼ 2) 웹스크래핑 기본 세팅

### 패키지 설치하기

```
beautifulsoup4, requests
```

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

### 스크래핑 코드 기본 세팅

```
import requests
from bs4 import BeautifulSoup

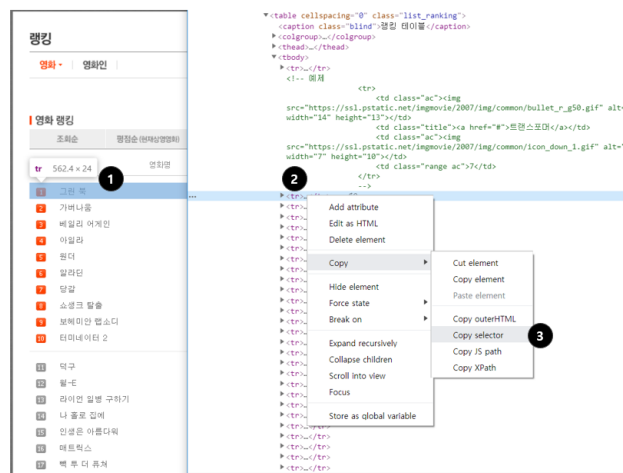
headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20190909', headers=headers)

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

#####
# (입맛에 맞게 코딩)
#####
```

## ▼ 3) beautifulsoup 라이브러리 내 select / select\_one 사용법

### 크롬 개발자도구의 Copy selector를 활용



select는 여러개가 리스트에 담겨서, select\_one은 한 개만 출력

```
# 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

## ▼ 4) 네이버영화 스크래핑 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/73.0.3683
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20190909', 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 not a_tag == None:
        # a의 text를 찍어본다.
        print(a_tag.text)
```

## ▼ 5) 엑셀 조작하기

### 패키지 설치하기

```
openpyxl
```

### 데이터를 읽어보기

```
from openpyxl import load_workbook

work_book = load_workbook('prac01.xlsx')
work_sheet = work_book['prac']

# 데이터를 읽어봅니다.
temp_text = work_sheet.cell(row = 1, column = 1).value

print(temp_text)
```

### 데이터를 입력하기

```
from openpyxl import load_workbook

work_book = load_workbook('prac01.xlsx')
work_sheet = work_book['prac']

# 데이터를 입력합니다.
work_sheet.cell(row=2, column=2, value='홍길동')

# 수정한 엑셀파일을 저장합니다.
# 참고: 다른이름으로 저장할 수도 있습니다.
work_book.save('prac01.xlsx')
```

## ▼ 6) mongoDB 조작하기

### 패키지 설치하기

```
pymongo
```

### DB연결하기

```
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.
```

### Insert 하기

```
# MongoDB에 insert 하기

doc = {
    'name': 'bobby',
    'age': 21
}

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

## Update 하기

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

db.users.update_one({'name': 'bobby'}, {'$set': {'age': 19}})
```

## Find 하기

```
# 여러개 뽑아보기
# 리스트형태. _id는 출력 안함
same_ages = list(db.users.find({'age': 21}, {'_id': 0}))
```

```
# 한 개 뽑아보기
# 딕셔너리형태. _id는 출력 안함
user = db.users.find_one({'name': 'bobby'}, {'_id': 0})
```

## ▼ 7) 네이버영화 스크래핑 → mongoDB 저장 코드

```
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
data = requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20190909', 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 not a_tag == None:
        title = a_tag.text
        star = movie.select_one('td.point').text

        doc = {
            'rank' : rank,
            'title' : title,
            'star' : star
        }
        db.movies.insert_one(doc)
        rank += 1
```

## ▼ 8) meta 태그 스크래핑하기

 meta태그 중 property가 og:image인 태그의 content 값을 가져오기

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

```
url = 'https://platum.kr/archives/120958'

headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683
data = requests.get(url,headers=headers)

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

og_image = soup.select_one('meta[property="og:image"]')

url_image = og_image['content']

print(url_image, url_title, url_description)
```

## API 만들기

### ▼ 1) Flask: 기본 구조+URL 라우팅

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return 'This is Home!'

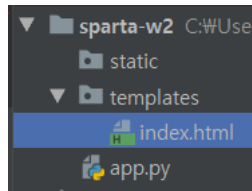
@app.route('/mypage')
def mypage():
    return 'This is My Page!'

if __name__ == '__main__':
    app.run('0.0.0.0',port=5000,debug=True)
```

### ▼ 2) Flask: html 파일 건네주기

- 기본 폴더구조

👉 flask 프레임워크를 쓸 때의 기본 폴더구조입니다. 오타에 유의하세요!  
만드시 templates 폴더내에 html 파일을 뒤야 불러올 수 있습니다.



- html 파일 불러오기

👉 render\_template를 import 에 추가하고, home()내에 써보았습니다.

```
from flask import Flask, render_template
app = Flask(__name__)

## URL 별로 함수명이 같거나,
## route('/') 등의 주소가 같으면 안됩니다.

@app.route('/')
def home():
    return render_template('index.html')

if __name__ == '__main__':
    app.run('0.0.0.0',port=5000,debug=True)
```

### ▼ 3) css, image가 깨질 때



👉 flask가 실행되는 위치를 기준으로 가져와야해서 그렇습니다.  
아래와 같이 바꿔주면, 깨지지 않습니다.

단, static 폴더 내 mystyle.css 파일을 위치시켜야 한다는 점!

```
<link rel="stylesheet" href="{{ url_for('static', filename='mystyle.css') }}">
```

#### ▼ 4) Flask: GET, POST API 만들기

👉 title\_give라는 키 값으로 들어온 데이터를 받는 방법입니다.

```
from flask import Flask, render_template, jsonify, request
app = Flask(__name__)

## HTML을 주는 부분
@app.route('/')
def home():
    return render_template('index.html')

## API 역할을 하는 부분
@app.route('/test', methods=['POST'])
def test_post():
    title_receive = request.form['title_give']
    print(title_receive)
    return jsonify({'result':'success', 'msg': '이 요청은 POST!'})

@app.route('/test', methods=['GET'])
def test_get():
    title_receive = request.args.get('title_give')
    print(title_receive)
    return jsonify({'result':'success', 'msg': '이 요청은 GET!'})

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

#### ▼ 5) Flask: 네이버영화 스크래핑 결과를 활용한 연습

##### ▼ pymongo 기본 세팅

```
from flask import Flask, render_template, jsonify, request
app = Flask(__name__)

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.

## HTML을 주는 부분
@app.route('/')
def home():
    return render_template('index.html')

(중략..)
```

##### ▼ GET연습: 받은 rank와 일치하는 영화 정보 가져오기

###### • API

👉 클라이언트가 준 값은, 모두 문자입니다. (즉, 숫자 2로 줘도 "2"로 인식)  
반면, DB에 저장된 rank의 값들은 숫자죠! robo3T에서 확인해볼까요?

그래서 비교하려면, 받은 rank\_give 값을 숫자로 꼭 바꿔줘야합니다!

```
@app.route('/test', methods=['GET'])
def test_get():
    # rank_give로 클라이언트가 준 rank를 가져오기
```

```

rank_receive = request.args.get('rank_give')

# rank_receive를 숫자로 만들어주기 (db엔 숫자로 저장되어있으니까!)
rank_receive = int(rank_receive)

# rank의 값이 받은 rank와 일치하는 document 찾기 & _id 값은 출력에서 제외하기
movie_info = db.movies.find_one({'rank':rank_receive},{'_id':0})

# info라는 키 값으로 영화정보 내려주기
return jsonify({'result':'success', 'info':movie_info})

```

- ajax로 잘 작동하는 지 확인

```

$.ajax({
  type: "GET",
  url: "/test?rank_give=2",
  data: {},
  success: function(response){
    console.log(response)
  }
})

```

#### ▼ POST연습: 받은 rank의 평점을 받은 star로 업데이트하기

- API

```

## API 역할을 하는 부분
@app.route('/test', methods=['POST'])
def test_post():
    # rank_give로 클라이언트가 준 rank를 가져오기 & 숫자변환
    rank_receive = request.form['rank_give']
    rank_receive = int(rank_receive)

    # star_give로 클라이언트가 준 star를 가져오기 & 숫자변환
    star_receive = request.form['star_give']
    star_receive = int(star_receive)

    # 해당 순위의 영화를 받은 score로 업데이트 해주기
    db.movies.update_one({'rank':rank_receive},{'$set':{'star':star_receive}})

    # 다했으면 성공여부만 보냄
    return jsonify({'result':'success'})

```

- ajax로 잘 작동하는 지 확인

```

$.ajax({
  type: "POST",
  url: "/test",
  data: {rank_give:2,star_give:5},
  success: function(response){
    console.log(response)
  }
})

```

#### ▼ POST연습: rank, title, star를 받아서 새로운 영화를 DB에 넣어주기

- API

```

@app.route('/new', methods=['POST'])
def new_post():
    rank_receive = int(request.form['rank_give'])
    star_receive = request.form['star_give']
    title_receive = request.form['title_give']

    db.movies.insert_one({'rank': rank_receive, 'star': star_receive, 'title':title_receive})

    return jsonify({'result': 'success'})

```

- ajax로 잘 작동하는 지 확인

```

$.ajax({
  type: "POST",
  url: "/new",

```

```

data: { rank_give:'99', star_give:'0', title_give:'청년경찰' },
success: function(response){
    console.log(response)
}
})

```

## ▼ 6) Flask: 나홀로메모장 전체 코드

```

from flask import Flask, render_template, jsonify, request
app = Flask(__name__)

import requests
from bs4 import BeautifulSoup

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

## HTML을 주는 부분
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/post', methods=['GET'])
def listing():
    # author_give 클라이언트가 준 author를 가져오기
    author_receive = request.args.get('author_give')
    # author의 값이 받은 author와 일치하는 document 찾기 & _id 값은 출력에서 제외하기
    result = list(db.articles.find({'author':author_receive},{ '_id':0}))
    # articles라는 키 값으로 영화정보 내려주기
    return jsonify({'result':'success', 'articles':result})

## API 역할을 하는 부분
@app.route('/post', methods=['POST'])
def saving():
    # 클라이언트로부터 데이터를 받는 부분
    url_receive = request.form['url_give']
    comment_receive = request.form['comment_give']
    author_receive = request.form['author_give']

    # meta tag를 스크래핑 하는 부분
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86'
    }
    data = requests.get(url_receive, headers=headers)

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

    og_image = soup.select_one('meta[property="og:image"]')
    og_title = soup.select_one('meta[property="og:title"]')
    og_description = soup.select_one('meta[property="og:description"]')

    url_image = og_image['content']
    url_title = og_title['content']
    url_description = og_description['content']

    # mongoDB에 넣는 부분
    article = {'author': author_receive, 'url': url_receive, 'comment': comment_receive, 'image': url_image,
               'title': url_title, 'desc': url_description}

    db.articles.insert_one(article)

    return jsonify({'result': 'success'})

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)

```

## 추가로 공부해보면 좋을 것들

### ▼ Flask - url에서 데이터를 받기

👉 필요한 경우, url의 일부 값을 변수로 가져올 수도 있습니다.

즉, 아래와 같이 입력한 상태에서,  
`http://localhost:5000/profile/sparta`

이라고 브라우저에 치면,  
`Hi! sparta`

라고 출력하게 됩니다.

```
@app.route("/profile/<username>")
def get_profile(username):
    return "Hi! " + username
```

#### ▼ Flask - render\_template으로 데이터를 주기

👉 render\_template으로 html 파일을 건네줄 때, 데이터를 미리 줄 수 도 있습니다.  
이 방법을 쓰면, 화면 로딩 후 ajax 요청을 하지 않아도 됩니다. (물론, 해도 됩니다.^^;)

아래와 같은 app.py, index.html을 만들어두고, 실행해보세요!  
`http://localhost:5000/sparta` 라고 치면,

Hi! 내 이름은 sparta입니다. 라고 나오게 될 것입니다.

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/<myname>')
def home(myname):
    return render_template('index.html', name=myname)

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

```
<html>
<head>
  <title>스파르타코딩클럽의 페이지</title>
  <!-- head 안에 들어 갈 어떤 요소들을 여기에 넣습니다 -->
</head>
<body>
  Hi! <h1>내 이름은 {{ name }}입니다.</h1>
</body>
</html>
```

[웹 개발할때 사용하기 좋은 무료 API가 정리된 깃헙\(클릭해서 이동\)>>](#)

## 파이썬으로 할 수 있는 다양한 것들

👉 "또 무얼 할 수 있죠?" 하는 분들을 위해 준비한 재미있는 것들!  
여기선 키워드만 알려드릴게요. 구글링을 통해 공부해보세요!

#### ▼ Selenium - 브라우저 제어

☞ 웹스크래핑을 해야하는데, 네이버 로그인을 꼭 해야한다면?

☞ 혼자 공부해보기: [\(링크1\)](#), ... 스스로 찾기!

#### ▼ 상세설명+코드 예시

- 패키지 설치하기

```
selenium
```

- chrome driver 설치하기

크롬 브라우저를 조작하려면, 제어해줄 수 있는 파일이 필요합니다. 아래 링크에서 드라이버를 다운 받습니다.

다운로드: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

- 1) 페이스북 창 열기

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

path = "C:/Users/bumky/Desktop/develop/chromedriver"
driver = webdriver.Chrome(path)
driver.get("http://www.facebook.org")
```

- 2) 페이스북에 로그인해보기

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

usr = '본인아이디'
pwd = '본인비밀번호'

path = "C:/Users/bumky/Desktop/develop/chromedriver"
driver = webdriver.Chrome(path)
driver.get("http://www.facebook.org")

assert "Facebook" in driver.title

elem = driver.find_element_by_id("email")
elem.send_keys(usr)
elem = driver.find_element_by_id("pass")
elem.send_keys(pwd)
elem.send_keys(Keys.RETURN)
```

- 3) 로그인하고, esc 눌러보기

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

usr = '본인아이디'
pwd = '본인비밀번호'

path = "C:/Users/bumky/Desktop/develop/chromedriver"
driver = webdriver.Chrome(path)
driver.get("http://www.facebook.org")

assert "Facebook" in driver.title

elem = driver.find_element_by_id("email")
elem.send_keys(usr)
elem = driver.find_element_by_id("pass")
elem.send_keys(pwd)
elem.send_keys(Keys.RETURN)

driver.implicitly_wait(5)
driver.find_element_by_tag_name('body').send_keys(Keys.ESCAPE)

driver.implicitly_wait(2)
```

- 4) esc 후에 포스팅 목록 불러오기

```
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

posts = soup.select('div.text_exposed_root')

for post in posts:
    print (post.text)
```

- 5) 10번 내리고, 포스팅 목록 불러오기

```
for i in range(10):
    driver.find_element_by_tag_name('body').send_keys(Keys.END)
    driver.implicitly_wait(10)
    time.sleep(5)
```

- 6) 완성된 코드

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

import time

from bs4 import BeautifulSoup

usr = '본인아이디'
pwd = '본인비밀번호'

path = "C:/Users/bumky/Desktop/develop/chromedriver"
driver = webdriver.Chrome(path)
driver.get("http://www.facebook.org")

assert "Facebook" in driver.title

elem = driver.find_element_by_id("email")
elem.send_keys(usr)
elem = driver.find_element_by_id("pass")
elem.send_keys(pwd)
elem.send_keys(Keys.RETURN)

driver.implicitly_wait(5)
driver.find_element_by_tag_name('body').send_keys(Keys.ESCAPE)

driver.implicitly_wait(5)
driver.find_element_by_tag_name('body').send_keys(Keys.ESCAPE)

for i in range(5):
    driver.find_element_by_tag_name('body').send_keys(Keys.END)
    driver.implicitly_wait(3)
    time.sleep(5)

html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

posts = soup.select('div.text_exposed_root')

for post in posts:
    print(post.text)
```

#### ▼ 셀레니움과 BeautifulSoup의 연결!



- 1) 셀레니움으로 브라우저를 열고,
- 2) HTML을 받아온 다음
- 3) BeautifulSoup으로 우리가 원하는 정보면 가지고 오면 가장 좋겠죠?

아래 코드를 참고해보세요. 빨간 음영 부분이 핵심입니다!

```
from selenium import webdriver
driver = webdriver.Chrome('C:\\Users\\bumky\\Desktop\\develop\\chromedriver')

from bs4 import BeautifulSoup
```

```
keyword = '코끼리'
url = 'https://www.google.com/search?q='+keyword+'&hl=ko&sxsrif=ACYBGNSTW5YFeVU0I4abA6H_bXsmwJ-gag:1582014089814&source=

driver.get(url)

req = driver.page_source

soup = BeautifulSoup(req, 'html.parser')

images = soup.select('#islr > div.islrc > div')

for count, image in enumerate(images):
    img = image.select_one('img')
    print(img['data-iurl'])
    if count == 5:
        break

driver.close()
```

## ▼ Telegram Bot - 텔레그램 봇

👉 텔레그램 봇을 쉽게 만들 수 있어요! (정말 쉬워요. 돈 워리!)

👉 혼자 공부해보기: [\(링크1\)](#), [\(링크2\)](#), ... 스스로 찾기!

## ▼ 코드 예시

```
from telegram.ext import Updater, MessageHandler, Filters # import modules

my_token = '950898370:AAEa9ZtiDT9beS0Gwb9QhHplqGtKgA0KEwo'

print('start telegram chat bot')

# message reply function
def get_message(bot, update):
    if '안녕' in update.message.text:
        update.message.reply_text('그래')
    elif '뭐해' in update.message.text:
        update.message.reply_text('그냥 있지')
    else:
        update.message.reply_text('뭐라는거야')

updater = Updater(my_token)

message_handler = MessageHandler(Filters.text, get_message)
updater.dispatcher.add_handler(message_handler)

updater.start_polling(timeout=3, clean=True)
updater.idle()
```

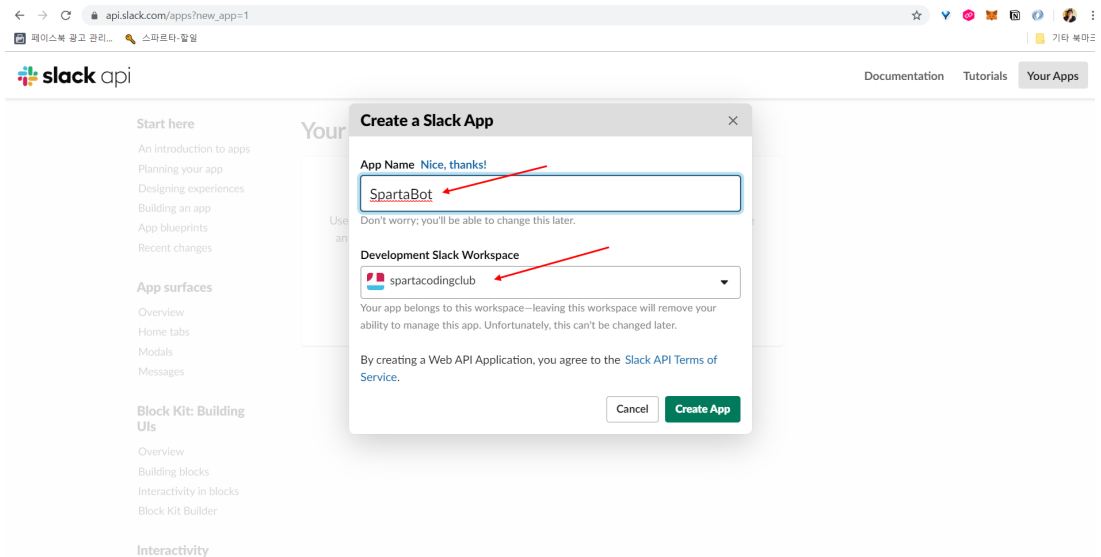
## ▼ Slack Bot - 슬랙 봇

👉 Telegram은 익숙하지 않다고요? 그럼 슬랙은 어떠신지요!

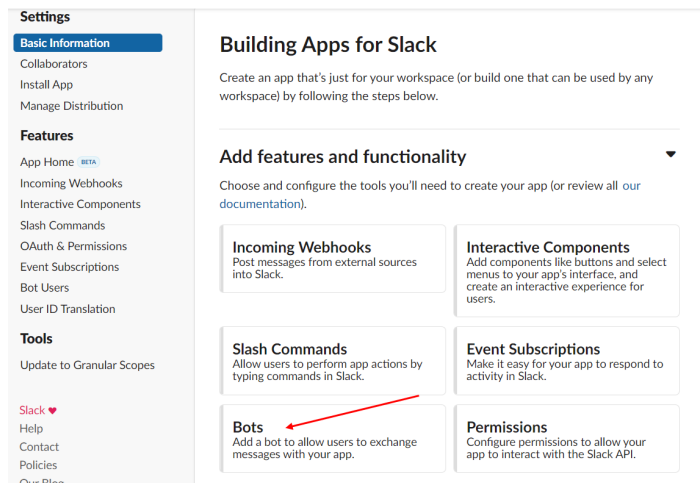
👉 혼자 공부해보기: [\(링크1\)](#), ... 스스로 찾기!

## ▼ 1) 토큰 발급받기

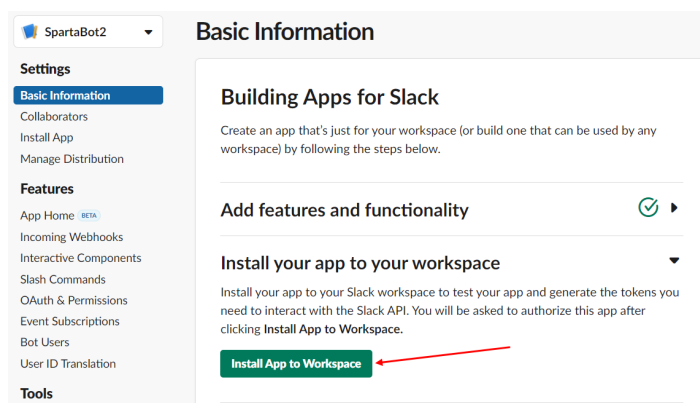
1. [Slack App 페이지](#)에 접속해서, Create an App를 클릭하고 봇 이름과 공유할 그룹을 입력



2. Basic Information에서, Bots를 클릭해 기능을 줍니다.

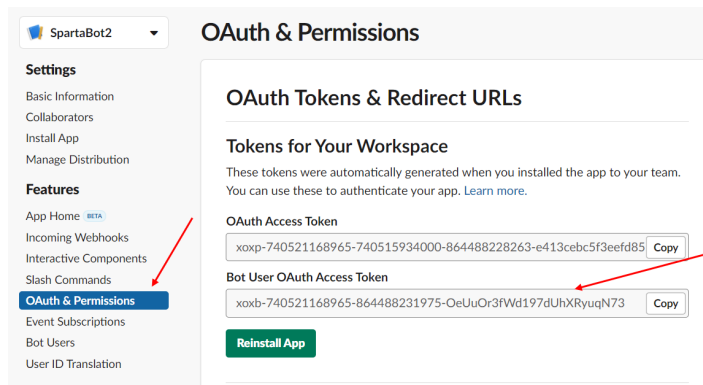


3. 다시 Basic Information에서, workspace에 앱(=봇)을 추가합니다.



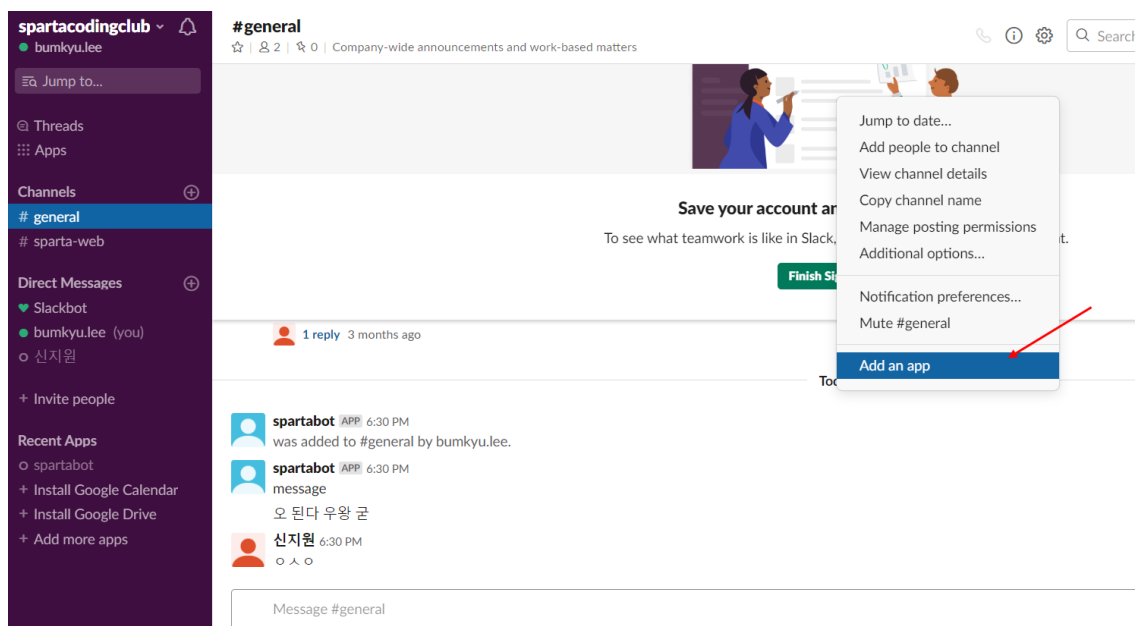
4. OAuth & Permissions에서 토큰을 복사해둡니다.





## ▼ 2) 봇을 팀 채널에 초대하기

슬랙 웹 또는 앱에서 Add an app을 클릭합니다.



Add를 클릭합니다.

## Add apps to # general

View App Directory

Search by name or category (e.g. productivity, sales)

### Add apps to get things done

Schedule meetings, share docs, and more — straight from Slack. [See app suggestions](#)



In your workspace (1)

SpartaBot2

Add

Add apps to your workspace



Google Drive

Get notifications about Google Drive files within Slack

Install

### ▼ 3) 봇이 메시지 보내게 하기



인스톨 할 패키지: slackclient

```
import slack
slack_token = 'xoxb-740521168965-864488231975-0eUu0r3fwd197duhXRyuqN73'
client = slack.WebClient(token=slack_token)
client.chat_postMessage(channel="#general", text="Hello world!")
```

### ▼ 4) 코드 예시 - 묻는 말에 대답하기



print(data) 로, 어떤 정보들을 불러올 수 있는지 확인해보세요  
chat\_postMessage 위에, 특정 기능을 수행하도록 코딩을 해둘 수 도 있습니다.  
(예를 들어 증권사API를 통해 주식을 사고 팔 수게 할 수도 있겠죠?)



slackclient 패키지를 인스톨하세요!

```
import slack

@slack.RTMClient.run_on(event='message')
def say_hello(**payload):
    data = payload['data']
    if 'bot_id' in data:
        return

    channel_id = data['channel']
    web_client = payload['web_client']

    chat = data.get('text', [])
    if 'hello' in chat:
        web_client.chat_postMessage(
            channel=channel_id,
            text='외국인이세요?'
        )
    elif '안녕' in chat:
```

# 받은메시지의 모든 정보  
# 봇이 보낸 메시지 일때는 패스

# 받은메시지의 채널 정보(어느 채널에서 왔는가?)  
# 받은메시지에 응답할 때 필요한 정보

# 받은메시지의 내용  
# 만약 chat에 hello라는 단어가 포함돼있으면  
# 답을 한다

# 만약 chat에 안녕이라는 단어가 포함돼있으면

```

web_client.chat_postMessage(          # 답을 한다
    channel=channel_id,
    text='한국인이세요?'
)
else:                                  # 이도저도 아니면
    web_client.chat_postMessage(      # 답을 한다
        channel=channel_id,
        text='무슨 말인지 모르겠어요!'
    )

slack_token = 'xoxb-740521168965-864488231975-0eUu0r3fwd197dUhxRyuqN73'
rtm_client = slack.RTMClient(token=slack_token)
rtm_client.start()

```

#### ▼ pyautogui - 키보드, 마우스 제어

👉 파이썬으로 키보드, 마우스를 제어할 수 있어요! → 매크로를 만들 수 있다는 사실!

👉 혼자 공부해보기: ([링크1](#)), ... 스스로 찾기!

#### ▼ gspread - 구글스프레드시트 연결

👉 구글스프레드 시트 연결도 물론 가능하죠! 읽기, 쓰기!

👉 혼자 공부해보기: ([링크1](#)), ... 스스로 찾기!

#### ▼ gmail 보내기 - 메일 보내기!

👉 안되는 게 없습니다! gmail, naver 메일 등 쉽게 보낼 수 있어요

👉 혼자 공부해보기: ([링크1](#)), ... 스스로 찾기!

##### 1. 파이썬으로 메일도 보내나요?!

네, 할 수 있습니다! 이처럼 누구나 많이 쓰는 동작들은 대부분 라이브러리(미리 짜여진 작은 프로그램)로 작성되어 있으니, 앞으로 여러분이 필요한 것이 있다면 구글에 검색해보세요! 거의 대부분 존재할 거예요 😎

##### 2. 기본 코드

```

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

me = "내 이메일 주소를 입력하세요."
my_password = "내 이메일 비밀번호를 입력하세요."
you = "상대방 이메일을 입력하세요"

## 여기서부터 코드를 작성하세요.
## 여기에서 코드 작성이 끝납니다.

# Gmail 관련 필요한 정보를 획득합니다.
s = smtplib.SMTP_SSL('smtp.gmail.com')
# Gmail에 로그인합니다.
s.login(me, my_password)
# 메일을 전송합니다.
s.sendmail(me, you, msg.as_string())
# 프로그램을 종료합니다.
s.quit()

```

##### 3. 완성 코드

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

me = "bumkyu.lee@gmail.com"
my_password = "vcpxrnjxegajhdn"
you = "bk.lee@spartacodingclub.kr"

## 여기서부터 코드를 작성하세요.
msg = MIMEMultipart('alternative')
msg['Subject'] = "Alert"
msg['From'] = me
msg['To'] = you

html = '<html><body><p>Hi, I have the following alerts for you!</p></body></html>'
part2 = MIMEText(html, 'html')

msg.attach(part2)
## 여기에서 코드 작성이 끝납니다.

# Send the message via gmail's regular server, over SSL - passwords are being sent, afterall
s = smtplib.SMTP_SSL('smtp.gmail.com')
s.login(me, my_password)
s.sendmail(me, you, msg.as_string())
s.quit()
```

#### ▼ [모르고 넘어가도 무방하지만, 궁금하면 살펴보세요! - SMTP, MIME란?]

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
```

? 위에서 쓰인 smtplib, mime 이런 것은 무엇인가요? 🤔

👉 이메일 전송 규약과 그에 관한 확장 프로토콜(약속)인 SMTP, MIME에 관한 라이브러리입니다. 이게 무슨 소리인가 하면요!

HTTP에 대해 들어보셨나요? HTTP가 우리가 일반적으로 아는 '웹'을 구성하는 프로토콜, 그러니까 하나의 약속입니다. HTML, CSS, Javascript가 무엇이고 그것을 어떻게 주고받을 것인지, 인증 처리는 어떻게 할 것인지에 대해 광범위한 규약이라고 할 수 있지요.

하지만 우리가 아는 '인터넷'은 오로지 HTTP로 구성되어있지 않습니다.

→ FTP(File Transfer Protocol): 파일을 송수신하기 위한 프로토콜

→ SMTP(Simple Mail Transfer Protocol): 글자로만 구성된 이메일을 주고받는 프로토콜

→ MIME(Multi-purpose Internet Mail Extensions): 이메일이 다른 종류의 데이터도 포함할 수 있도록 도와주는 프로토콜

정리하자면, 이메일과 관련해서는!

글자로만 구성된 이메일을 주고받기 위해 반드시 지켜야 하는 규칙인 SMTP와 더불어, 이메일에 글자 뿐만 아니라 이미지, 동영상 등을 주고받을 수 있도록 도와주는 규칙인 MIME이 존재합니다. 이 규칙을 준수하기 위해 코드 상단에 위 세 줄이 포함된 것이구요.

더 자세한 내용은 다음 링크를 살펴보세요.

([링크 1](#)), ([링크 2](#)), ...

#### ▼ PyJWT - 로그인 구현하기

👉 편의를 위해, 1~4주차에서 배운 수준으로 로그인을 구현해두었습니다.  
아래 깃 주소를 clone 받아 코드를 살펴봐주세요!

주석으로 상세 설명을 적어두었습니다.  
[https://github.com/bumkyulee/login\\_prac](https://github.com/bumkyulee/login_prac)

👉 회원가입은 id / pw 를 받아 DB에 저장하고,  
로그인은 id / pw 를 받아 DB에 있는 것과 맞춰보는 과정이지만,

로그인하고 요청하는 API들(예 - 회원정보 확인, 내 글 목록 확인 등)에서는  
어떻게 이 API가 로그인된 회원에게서 온 것이라고 확인할 수 있을까요?

로그인은 크게 아래와 같은 방법으로 구현된답니다.  
토큰을 만드는 방법은 여러가지가 있는데, JWT([링크](#))를 이용하면 편리하답니다.

스파르타의 예시 코드를 살펴봐주세요!

1. **회원가입** : id / pw 를 받아서 DB에 저장한다. 단, pw는 암호화해서 저장한다.
2. **로그인** : id / pw를 받아서, pw를 암호화 시킨 다음에 DB에 있는 정보와 비교한다. 성공하면 '몇시간짜리 토큰'을 준다.
3. **로그인 이후**: 모든 API를 요청할 때, '몇시간짜리 토큰'을 같이 준다. 서버에서는 동작을 수행하기 전에, 해당 토큰이 유효한 것인지를 판단하고, API의 본래 역할을 수행한다.

👉 혼자 공부해보기: ([링크1](#)), ([링크2](#)), ([링크3](#)) ... 스스로 찾기!

#### ▼ Schedule - 주기적으로 실행하기

👉 주기적으로 함수를 실행해야 한다면? (매일 아침 10시 or 10초마다 등)  
schedule 라이브러리를 활용해보세요!

👉 혼자 공부해보기: ([링크1](#)), ... 스스로 찾기!

#### ▼ 코드

```
import schedule

def job():
    print('여기에 할 일을 넣기')

def run():
    schedule.every().day.at('09:00').do(job) # 매일 09:00 마다 job 함수를 실행
    while True:
        schedule.run_pending()

if __name__ == "__main__":
    run()
```

... and more! 상상할 수 있는 모든 것 모두!