

Bioinformatics Web Programming

생물정보학 웹 프로그래밍

한주현, 2019.08

```
<html>  
Hello BioInfomatics  
</html>
```



목차

1. 강사 소개
2. Bioinformatics Web Programming?

3. 기초 쌓기
 - 개발환경 만들기
 - vscode 설치하기
 - 기본기 - HTML, CSS, Javascript
 - index.html 만들기
 - css로 스타일 적용하기
 - javascript로 웹페이지 프로그래밍 하기
 - 개발을 더 쉽게 - Bootstrap, JQuery
 - Bootstrap 살펴보기
 - JQuery 적용하기

목차

4. 생물정보학 웹 프로그래밍

- DNA Counter
- GC(%) 계산기
- Reverse Complementary Sequence 생성기

5. 고급 웹 프로그래밍

- Web framework
- Express.js

강사 소개

강사 소개	
이름	한주현
소속	전) 마크로젠 데이터분석부 현) 쓰리빌리언 Bioinformatics Engineer 현) 서울대학교 의료정보학 전공 이규언 교수님 연구실
메일	kenneth.jh.han@snu.ac.kr
주요 업무	Rare disease genetic algorithms Bioinformatics tool development Human genome analysis (WGS, WES) Analysis pipeline / platform Development Full stack development
주 언어	Python, JAVA, JavaScript, Bash shell
저서	니콜라스 볼커 이야기 (2016.10, 금창원 외 공역) 바이오파이썬으로 만나는 생물정보학 (2019.03, 한주현)
강의	바이오협회 유전체 예비 전문가 과정 3-7기 서울여자대학교 초청 강의
웹 페이지	https://korbillgates.tistory.com (블로그)



Bioinformatics Web Programming

- 생물정보학 웹 프로그래밍
- 누구를 위한 강의?
 - 웹 프로그래밍을 하고 싶은 의학, 생명과학 관련 전공자
 - 생물정보학 지식을 웹 프로그래밍에 접목 시키고 싶은 개발자
 - 간단한 웹페이지 부터 API 호출 및 서버와 통신하는 웹 어플리케이션을 제작하고 싶은 사람
- 선행 지식
 - 기본적인 생물학 지식 (고교 수준 분자생물학)
 - 기본적인 프로그래밍 지식
 - HTML, CSS, JavaScript은 몰라도 OK!!

고민들

- 사실 웹이란 지식이 쌓아야 할 배경이 꽤나 많습니다.
- HTML, CSS, JavaScript 가 기본이며, BootStrap, jQuery를 다룰 줄 알면 쉽게 페이지를 만들 수 있습니다. (front end)
- 또한 이렇게 만들면 html 페이지 그 자체 밖에 만들 수 없기에 데이터베이스와 연결 또는 서버의 계산이 필요한 경우 서버쪽 요소를 고려하여 서버와 통신을 하는 페이지를 만들 수 있어야 합니다. (back end)
- 여러가지 언어가 있지만 우리는 JavaScript 기반으로 프로그래밍을 진행해보겠습니다.
- Back end 단을 nodejs로 구성해보려합니다.
- 그리고 Front end 단에서도 많은 발전이 이루어져 AngularJs, ReactJs, VueJs 등이 있습니다만 이번 강좌에서는 다루지 않겠습니다. 관심이 있으시다면 앞으로의 학습방향으로 설정하시면 좋겠습니다.



우리가 만들 목표

A screenshot of a web browser showing the homepage of "Bioinformatics Web Programming". The title "Bioinformatics Web Programming" is displayed prominently in the center of a blue-toned background featuring a glowing DNA helix and a network of interconnected nodes. Below the title, there are three main sections: "About", "Purpose", and "Details". The "About" section includes a brief description and a link to a lecture by Joohyun Han. The "Purpose" section describes the learning objective as "Learning by Making Bioinformatics Web Program with Bioinformatics Examples". The "Details" section lists the technologies used, such as HTML, CSS, JavaScript, and Python.

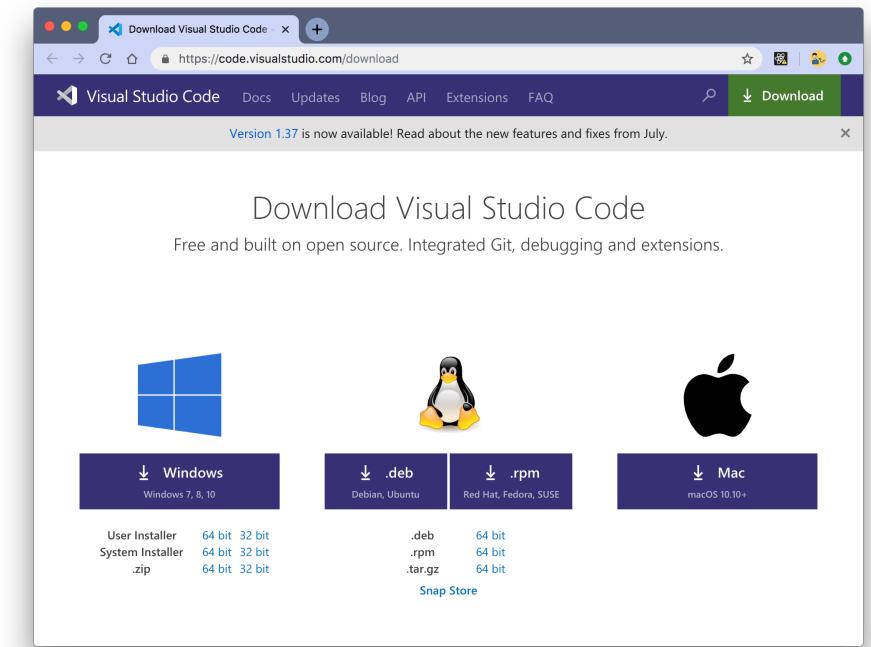
우리가 만들 목표를 미리 보고 가자!!



개발 환경 구축하기

- Visual Studio Code

- ✓ Microsoft에서 2016년에 만든 텍스트 에디터
- ✓ 윈도우, 리눅스, 맥을 지원
- ✓ 방대한 확장 기능
- ✓ 여러 프로그래밍 언어를 지원
- ✓ 웹 개발 시 매우 편함



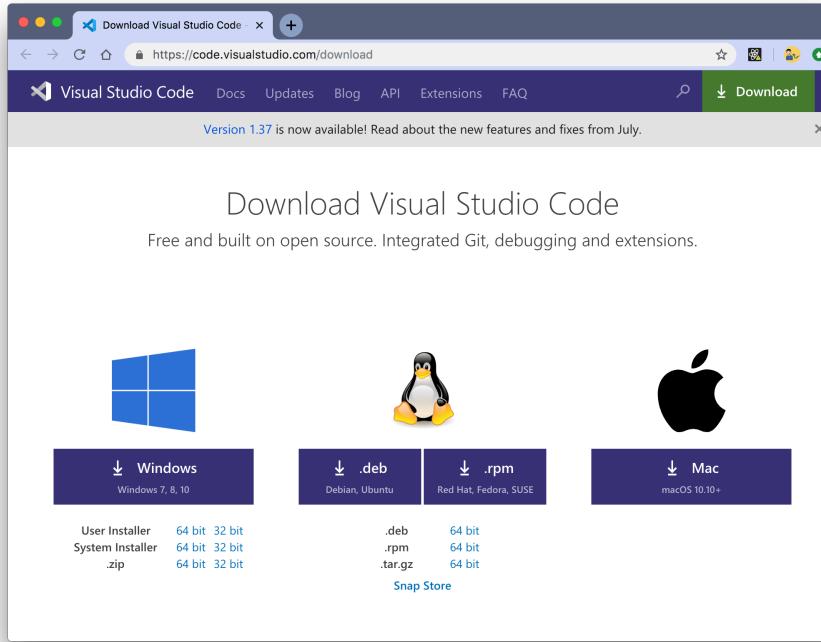
<https://code.visualstudio.com/download>

개발 환경 구축하기

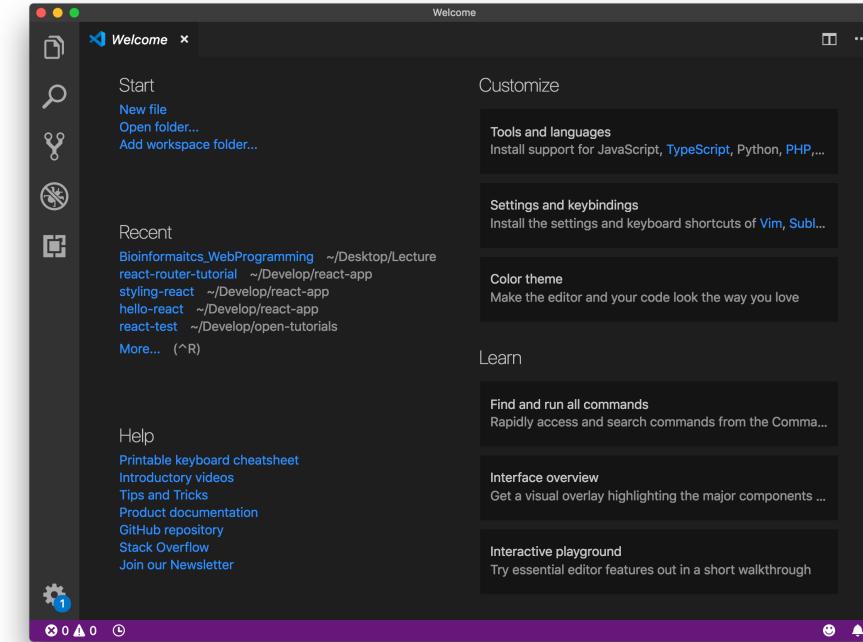
진행이 어려우신 분들은
언제든 손들어 주세요~



- 운영체제에 맞게 Visual studio code를 설치해봅시다!



<https://code.visualstudio.com/download>



설치 후 실행 화면

개발 환경 구축하기

- BioInfoWeb 디렉토리를 만들고
- Visual Studio Code 에서 디렉토리를 엽니다.
- html 이라는 디렉토리를 만듭니다.

데이터 받아오기

BioInfoWeb 디렉토리에 다음 github 주소의 내용을 받습니다

https://github.com/KennethJHan/bioinformatics_web_programming

git clone
을 사용하면 편하다



01 HTML

Hyper **T**ext **M**ark-up **L**anguage

태그(Tag)를 사용하여 웹페이지의 구조를 만드는 마크업 언어다.

HTML 태그의 구성요소

< a href="hello.html" > 안녕 </ a >

속성 이름 속성 값
attribute name attribute value

속성
attribute

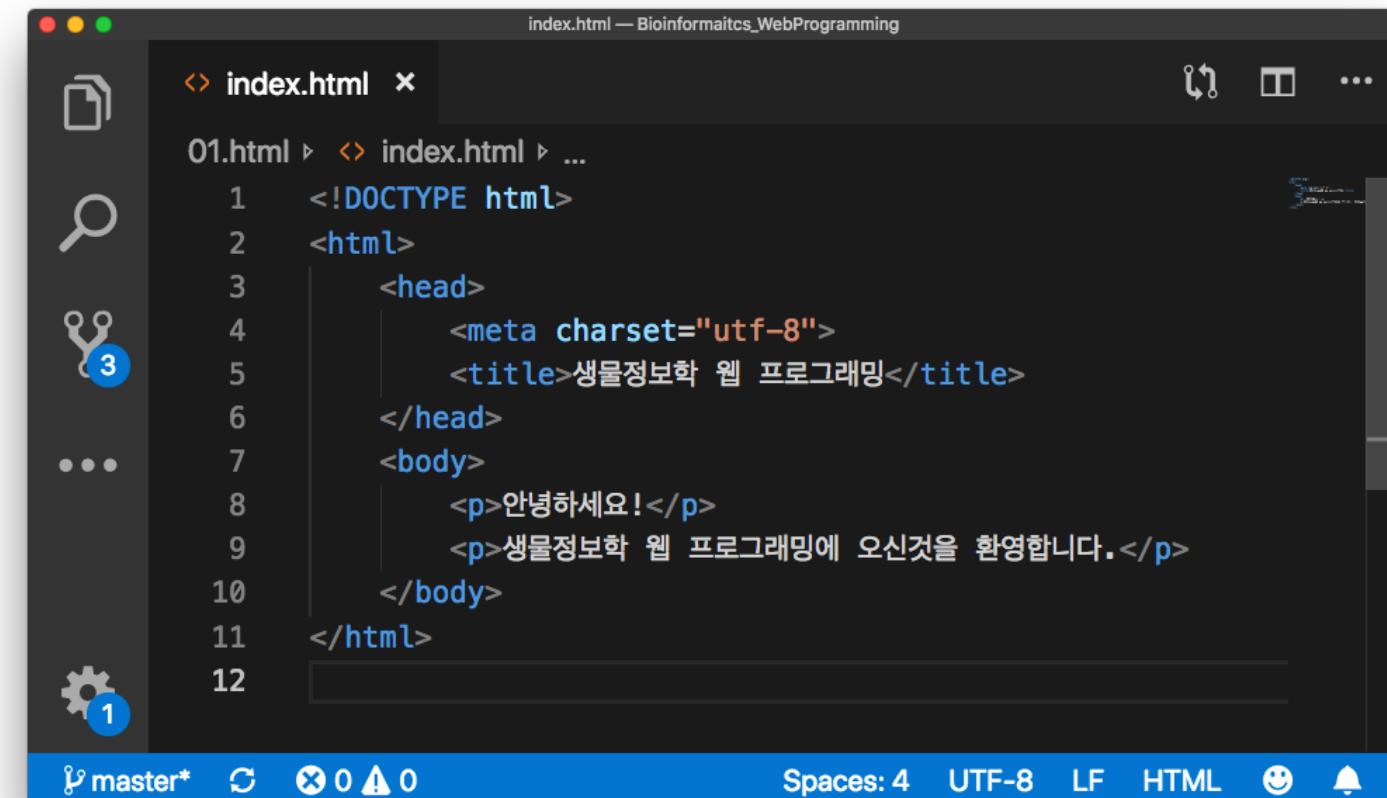
시작 태그

끝 태그

우선 다음 슬라이드에서
타이핑 후 자세한 내용을
알아보자!



다음과 같이
index.html 이라는 파일에 타이핑 후
저장합니다.

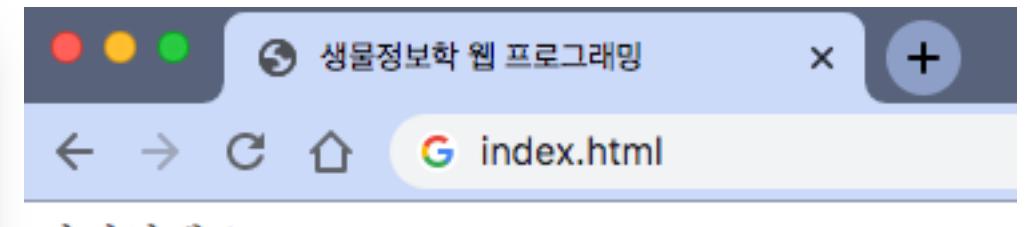


```
index.html — Bioinformatics_WebProgramming
index.html ×

01.html > index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>생물정보학 웹 프로그래밍</title>
6      </head>
7      <body>
8          <p>안녕하세요!</p>
9          <p>생물정보학 웹 프로그래밍에 오신것을 환영합니다.</p>
10     </body>
11 </html>
12
```

Spaces: 4 UTF-8 LF HTML ☺ 🔔

브라우저를 엽니다. (크롬 or 파이어폭스 추천)
탐색기에서 저장한 파일을 브라우저로
드래그 & 드롭 합니다.



안녕하세요!

생물정보학 웹 프로그래밍에 오신것을 환영합니다.

우와..
넘 신기하다



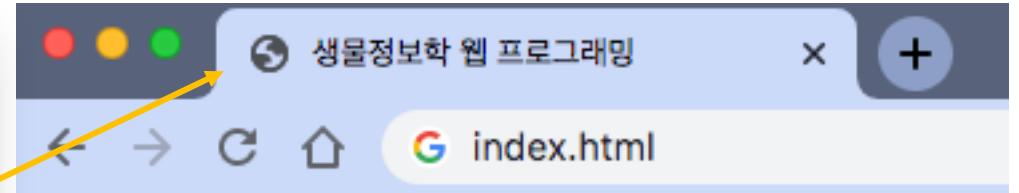
HTML5 문서임을 알려줌

```
index.html — Bioinformatics_WebProgramming
index.html ×
01.html ▶ index.html ▶ ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>생물정보학 웹 프로그래밍</title>
6      </head>
7      <body>
8          <p>안녕하세요!</p>
9          <p>생물정보학 웹 프로그래밍에 오신것을 환영합니다.</p>
10     </body>
11 </html>
12
```

html 문서는
<html> 태그로
감싼다

html 문서의
속성을 지정하는 곳
<meta>, <title> 등이 들어감

html 문서의
본문을 나타내는 곳

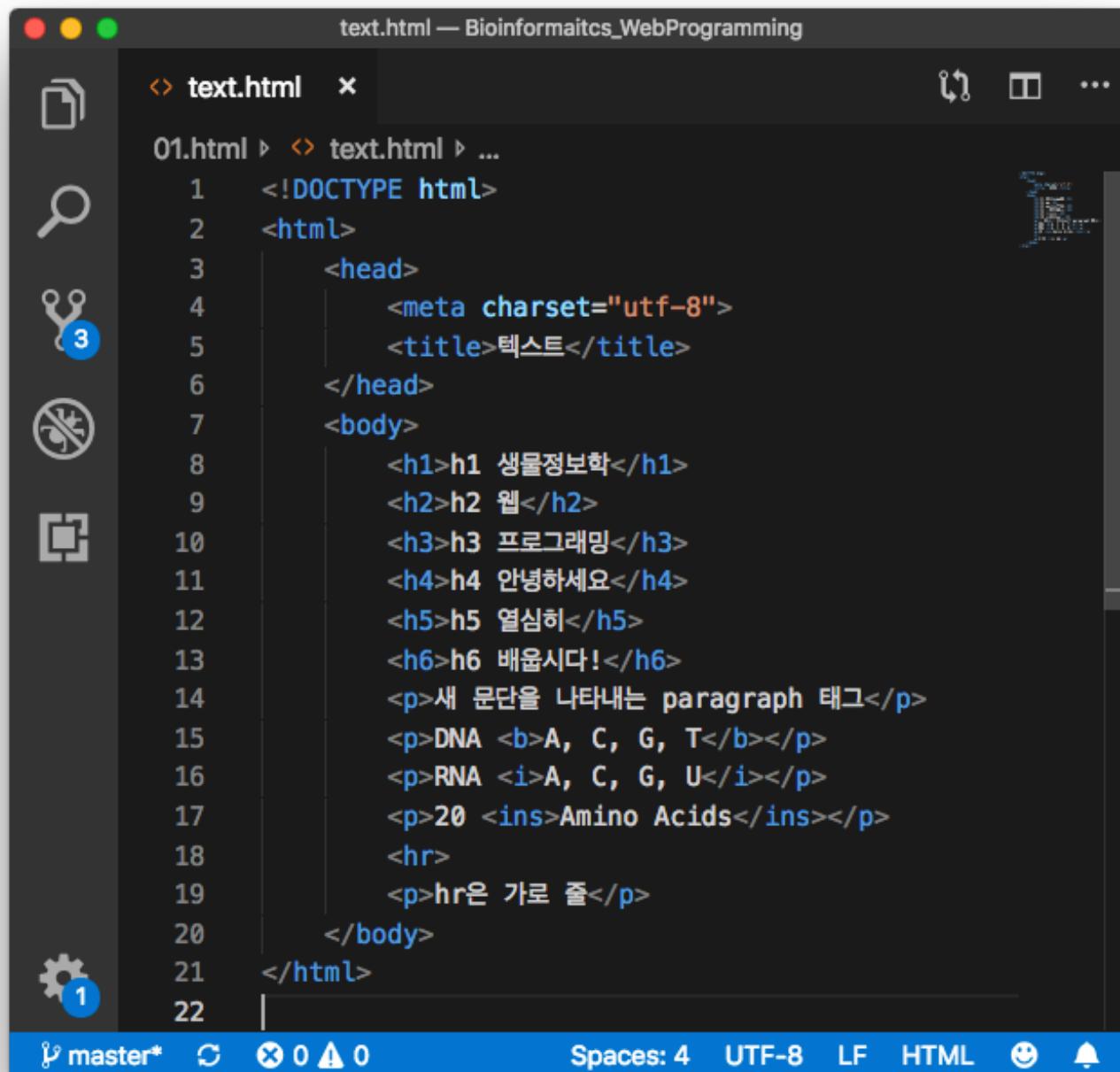


안녕하세요!
생물정보학 웹 프로그래밍에 오신것을 환영합니다.

대부분의
HTML 문서는
이러한 형식이다.



태그의 종류 - 텍스트

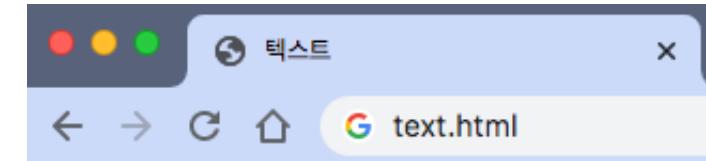


text.html — Bioinformatics_WebProgramming

01.html > text.html > ...

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>텍스트</title>
6      </head>
7      <body>
8          <h1>h1 생물정보학</h1>
9          <h2>h2 웹</h2>
10         <h3>h3 프로그래밍</h3>
11         <h4>h4 안녕하세요</h4>
12         <h5>h5 열심히</h5>
13         <h6>h6 배웁시다!</h6>
14         <p>새 문단을 나타내는 paragraph 태그</p>
15         <p>DNA <b>A, C, G, T</b></p>
16         <p>RNA <i>A, C, G, U</i></p>
17         <p>20 <ins>Amino Acids</ins></p>
18         <hr>
19         <p>hr은 가로 줄</p>
20     </body>
21 </html>
```

Spaces: 4 UTF-8 LF HTML ⚡ 🔔

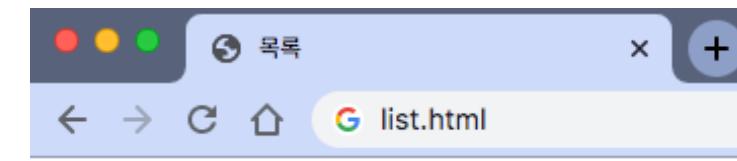


태그의 종류 - 목록

The screenshot shows a dark-themed code editor with a sidebar containing icons for file operations, search, and settings. The main area displays the content of 'list.html'. The code includes a title '목록', an unordered list of three items (한주현, 바이오파이썬으로 만나는 생물정보학, 생물정보학 웹 프로그래밍), and an ordered list of three items (Agarose gel 만들기, Sample 로딩, 결과 확인). The code uses standard HTML tags like and .

```
list.html — Bioinformatics_WebProgramming
text.html    list.html ×
01.html > list.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>목록</title>
6      </head>
7      <body>
8          <p>생물정보학 (순서가 없는 목록, unordered list)</p>
9          <ul>
10             <li>한주현</li>
11             <li>바이오파이썬으로 만나는 생물정보학</li>
12             <li>생물정보학 웹 프로그래밍</li>
13         </ul>
14
15         <p>전기영동 순서 (순서가 있는 목록, ordered list)</p>
16         <ol>
17             <li>Agarose gel 만들기</li>
18             <li>Sample 로딩</li>
19             <li>결과 확인</li>
20         </ol>
21     </body>
22 </html>
23
```

master* 0 0 0 0 1 Spaces: 4 UTF-8 LF HTML



생물정보학 (순서가 없는 목록, unordered list)

- 한주현
- 바이오파이썬으로 만나는 생물정보학
- 생물정보학 웹 프로그래밍

전기영동 순서 (순서가 있는 목록, ordered list)

1. Agarose gel 만들기
2. Sample 로딩
3. 결과 확인

태그의 종류 - 링크, 그림

link_image.html — Bioinformatics_WebProgramming

link_image.html x

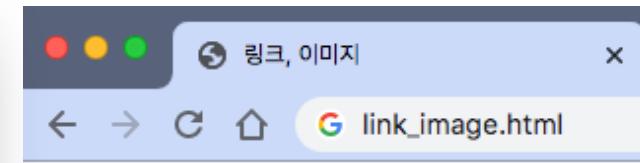
01.html > link_image.html > ...

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>링크, 이미지</title>
6      </head>
7      <body>
8          <div>
9              <a href="index.html">index</a>
10         </div>
11         <div>
12             <a href="https://ko.wikipedia.org/wiki/헤모글로빈">
13                 
14             </a>
15         </div>
16     </body>
17 </html>
```

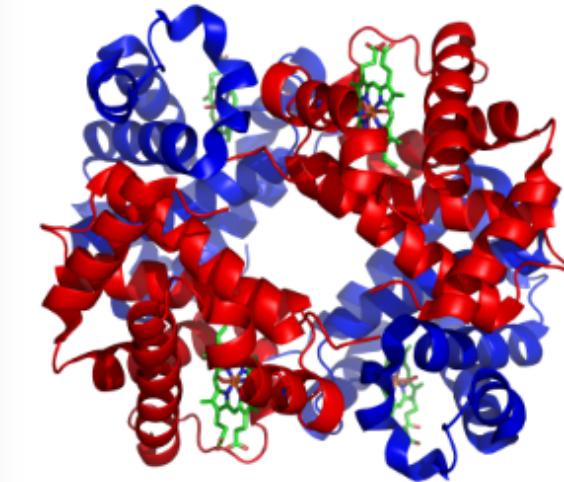
Spaces: 4 UTF-8 LF HTML

그림 출처

https://en.wikipedia.org/wiki/Hemoglobin#/media/File:1GZX_Haemoglobin.png



index



그림은

html_css_js_bootstrap_jquery
디렉토리에 있으니 본인 작업 딕렉
토리에 복사하여 사용하자!

링크와 그림을 클릭해보자!



태그의 종류 - 테이블

A screenshot of a code editor window titled "table.html — Bioinformatics_WebProgramming". The editor shows the following HTML code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>테이블</title>
6      </head>
7      <body>
8          <p>뉴클레오티드</p>
9          <table>
10             <tr>
11                 <th>문자</th>
12                 <th>설명</th>
13                 <th>계열</th>
14             </tr>
15             <tr>
16                 <td>A</td>
17                 <td>Adenine</td>
18                 <td>Purine</td>
19             </tr>
20             <tr>
21                 <td>C</td>
22                 <td>Cytosine</td>
23                 <td>Pyrimidine</td>
24             </tr>
25             <tr>
26                 <td>G</td>
27                 <td>Guanine</td>
28                 <td>Purine</td>
29             </tr>
30             <tr>
31                 <td>T</td>
32                 <td>Thymine</td>
33                 <td>Pyrimidine</td>
34             </tr>
35             <tr>
36                 <td>U</td>
37                 <td>Uracil</td>
38                 <td>Pyrimidine</td>
39             </tr>
40         </table>
41     </body>
42 </html>
```

The status bar at the bottom indicates "master*" and "Spaces: 4 UTF-8 LF HTML".

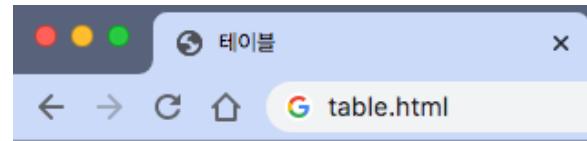


table row안에
table header와
table data가
들어있다!



태그의 종류 - 폼

form.html — Bioinformatics_WebProgramming

form.html

```
01.html > form.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>폼</title>
6      </head>
7      <body>
8          <h2>입력 폼</h2>
9          <form action="">
10             <p>서열</p>
11             <input type="text" name="seq">
12             <p>서열 종류</p>
13             <input type="radio" name="seqtype" value="DNA">DNA<br>
14             <input type="radio" name="seqtype" value="RNA">RNA<br>
15             <input type="radio" name="seqtype" value="Protein">Protein<br>
16             <input type="submit" value="제출">
17         </form>
18     </body>
19 </html>
```

master* 0 0 0 3 html | form.html Ln 20, Col 1 Spaces:

폼

← → ⌂ ⌂ G form.html?seq=ACGT&seqtype=DNA

입력 폼

서열

ACGT

서열 종류

DNA (checked)
RNA
Protein

제출

form 안에 들어있는 input 들을 사용자로 부터 받아 처리한다.

만든 뒤 제출 버튼을 눌러보자!
무엇이 바뀌었을까? (힌트 주소창)

form 처리 내용은 후에 다룬다!



02 CSS

- Cascading Style Sheet
- 구조를 나타내는 HTML에 첨가하는 디자인 정보를 담고 있는 언어
- css라는 확장자를 사용한 외부 파일을 불러온다

```
body {  
    background-color: lightblue;  
}
```

멋진 디자인이 들어간
웹페이지 제작을 위해
CSS를 사용하여 꾸민다!

기존에 만든 페이지에 CSS를
적용해보자!



CSS 적용

The screenshot shows a code editor with two tabs: 'text.html' and '# text.css'. The 'text.html' file contains an HTML document with various headings and paragraphs. The '# text.css' file contains a CSS stylesheet with rules for the body, h1, and p elements.

```
text.html — Bioinformatics_WebProgramming
text.html
01.html > text.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>텍스트</title>
6          <link rel="stylesheet" href="text.css">
7      </head>
8      <body>
9          <h1>Bioinformatics</h1>
10         <h2>Web</h2>
11         <h3>Programming</h3>
12         <h4>h4 안녕하세요</h4>
13         <h5>h5 열심히</h5>
14         <h6>h6 배웁시다!</h6>
15         <p>새 문단을 나타내는 paragraph 태그</p>
16         <p>DNA <b>A, C, G, T</b></p>
17         <p>RNA <i>A, C, G, U</i></p>
18         <p>20 <ins>Amino Acids</ins></p>
19         <hr>
20         <p>Bioinformatics Web Programming</p>
21     </body>
22 </html>
23
```

```
# text.css
01.html > # text.css > ...
1  body {
2      background-color: #lightgray;
3  }
4  h1 {
5      background-color: #3C5A99;
6      color: white;
7      text-align: center;
8  }
9  p {
10     color: #rgb(200, 100, 0);
11 }
```

The screenshot shows a web browser window titled '텍스트' displaying the rendered HTML from 'text.html'. The page has a dark blue header with 'Bioinformatics' and 'Web Programming'. Below the header, there are several sections: 'h4 안녕하세요', 'h5 열심히', 'h6 배웁시다!', a paragraph about DNA/RNA, and a paragraph about amino acids. At the bottom, there is a footer with the text 'Bioinformatics Web Programming'.

Bioinformatics

Web

Programming

h4 안녕하세요

h5 열심히

h6 배웁시다!

새 문단을 나타내는 paragraph 태그

DNA A, C, G, T

RNA A, C, G, U

20 Amino Acids

Bioinformatics Web Programming

CSS 적용

The image shows a code editor window with two tabs: "table.html" and "table.css".

table.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>테이블</title>
    <link rel="stylesheet" href="table.css">
  </head>
  <body>
    <p class="title">Nucleotide</p>
    <table>
      <tr>
        <th>문자</th>
        <th>설명</th>
        <th>계열</th>
      </tr>
      <tr>
        <td>A</td>
        <td>Adenine</td>
        <td>Purine</td>
      </tr>
      <tr>
        <td>C</td>
        <td>Cytosine</td>
        <td>Pyrimidine</td>
      </tr>
      <tr>
        <td>G</td>
        <td>Guanine</td>
        <td>Purine</td>
      </tr>
      <tr>
        <td>T</td>
        <td>Thymine</td>
        <td>Pyrimidine</td>
      </tr>
      <tr>
        <td>U</td>
        <td>Uracil</td>
        <td>Pyrimidine</td>
      </tr>
    </table>
  </body>
</html>
```

table.css:

```
table {
  border-collapse: collapse;
  width: 100%;
}

th {
  background-color: #4285F4;
  color: white;
}

th, td {
  text-align: left;
  padding: 8px;
}

tr:nth-child(even) {
  background-color: #f2f2f2;
}

.title {
  text-align: center;
  font-weight: bold;
  font-size: 30px;
}
```

Bottom status bar: Ln 25, Col 1 Spaces: 4 UTF-8 LF CSS

The image shows a web browser window displaying a table titled "Nucleotide".

Nucleotide

문자	설명	계열
A	Adenine	Purine
C	Cytosine	Pyrimidine
G	Guanine	Purine
T	Thymine	Pyrimidine
U	Uracil	Pyrimidine

A yellow speech bubble contains the Korean text "멋지다!" (Nice!). A cartoon green snake is also present.

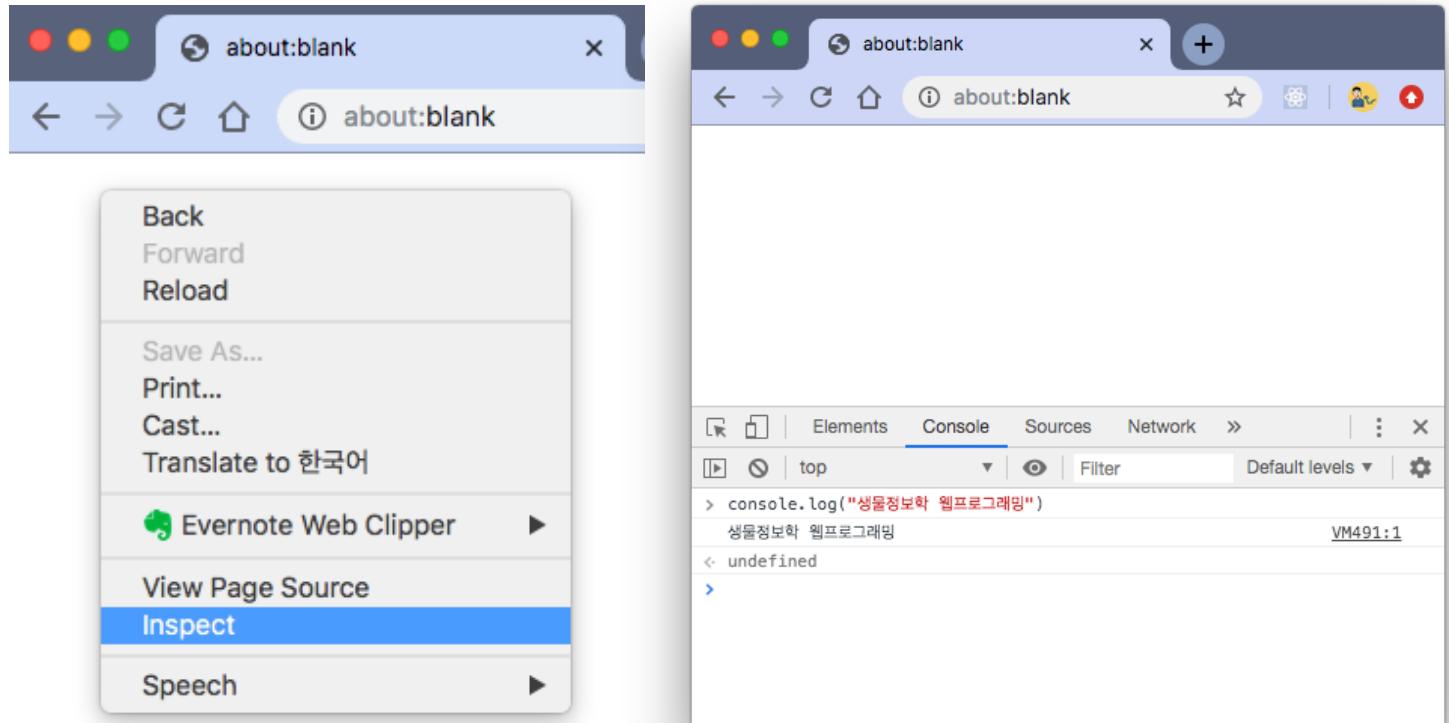
03 JavaScript

- "HTML: 구조, CSS: 디자인" 이라 할 때 JavaScript는 웹페이지의 동작의 역할을 맡고 있다.
- 프로그래밍 언어이며 스크립트형 언어다.
- 웹과는 땄래야 땄 수 없는 중요한 언어
- 구글의 V8 JavaScript 엔진의 개발로 비약적으로 속도가 증가 (크롬을 쓰자!!)
- AJAX(비동기식 정보 교환),
jQuery(클라이언트 사이드 단순화),
Node.js(서버사이드 js 프로그래밍),
AngularJS, VueJS, ReactJS(프론트엔드),
와 같은 기라성 같은 역작들로 웹세계를 평정했다.

JAVA와 JavaScript는
프로그래밍 언어라는 공통점 빼곤
!아!무!런! 관계가 없다.
마치..
인도 : 인도네시아
햄 : 햄스터
처럼..



Inspect와 console.log()



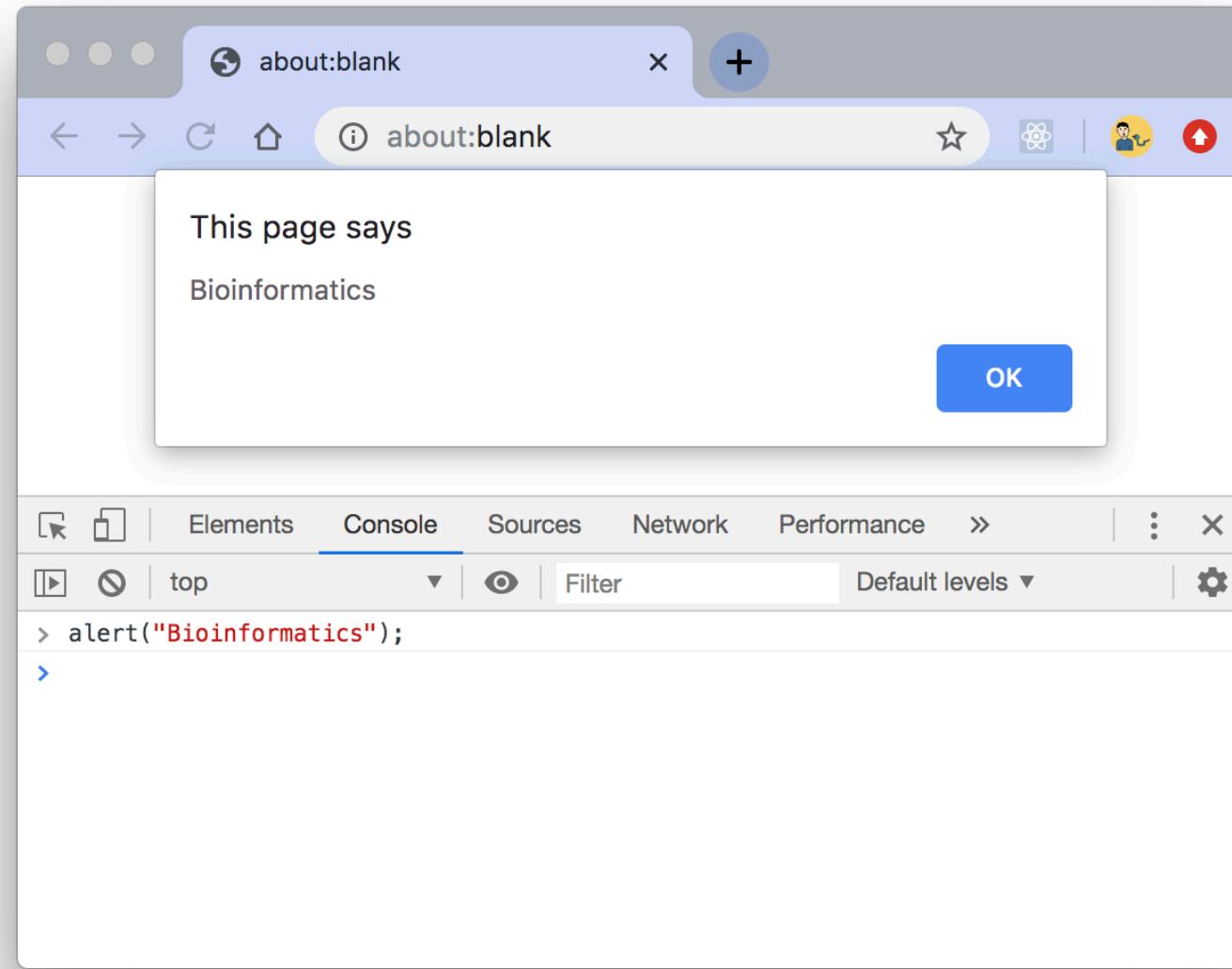
- JavaScript 연습장

크롬의 주소창에 `about:blank` 를 타이핑 하여 들어간다
빈 화면에서 우클릭 후 `Inspect`를 클릭한다
탭 중 `Console`에 들어간다

console.log() 는
() 내부의 내용을 출력해주는 메서드로
JavaScript로 작성한 프로그램을
디버깅 할 때 유용하게 사용할 수 있다



alert()



console.log() 와 함께
내용을 출력 할 수 있는 방법 중
하나로
개발자 뿐만 아니라 사용자에게
메시지를 전달할 수 있는 방법!



JavaScript 기초

- JavaScript 변수 선언

```
> var num1 = 10;  
> var num2 = 20;  
> var num3 = num1 + num2;  
> console.log(num3);  
30
```

- JavaScript 배열

```
> var arr1 = ["APOB", "BRCA1", "EGFR"];  
> for(i=0; arr1.length; i++) {  
    console.log(arr1[i]);  
}  
APOB  
BRCA1  
EGFR
```

- JavaScript 문자열

```
> var seq1 = "AGT";  
> var seq2 = "TAG";  
> var seq3 = seq1 + seq2;  
> console.log(seq3);  
AGTTAG
```

- JavaScript 주석달기

```
// JavaScript 주석  
/*  
여러 줄  
주석  
*/
```

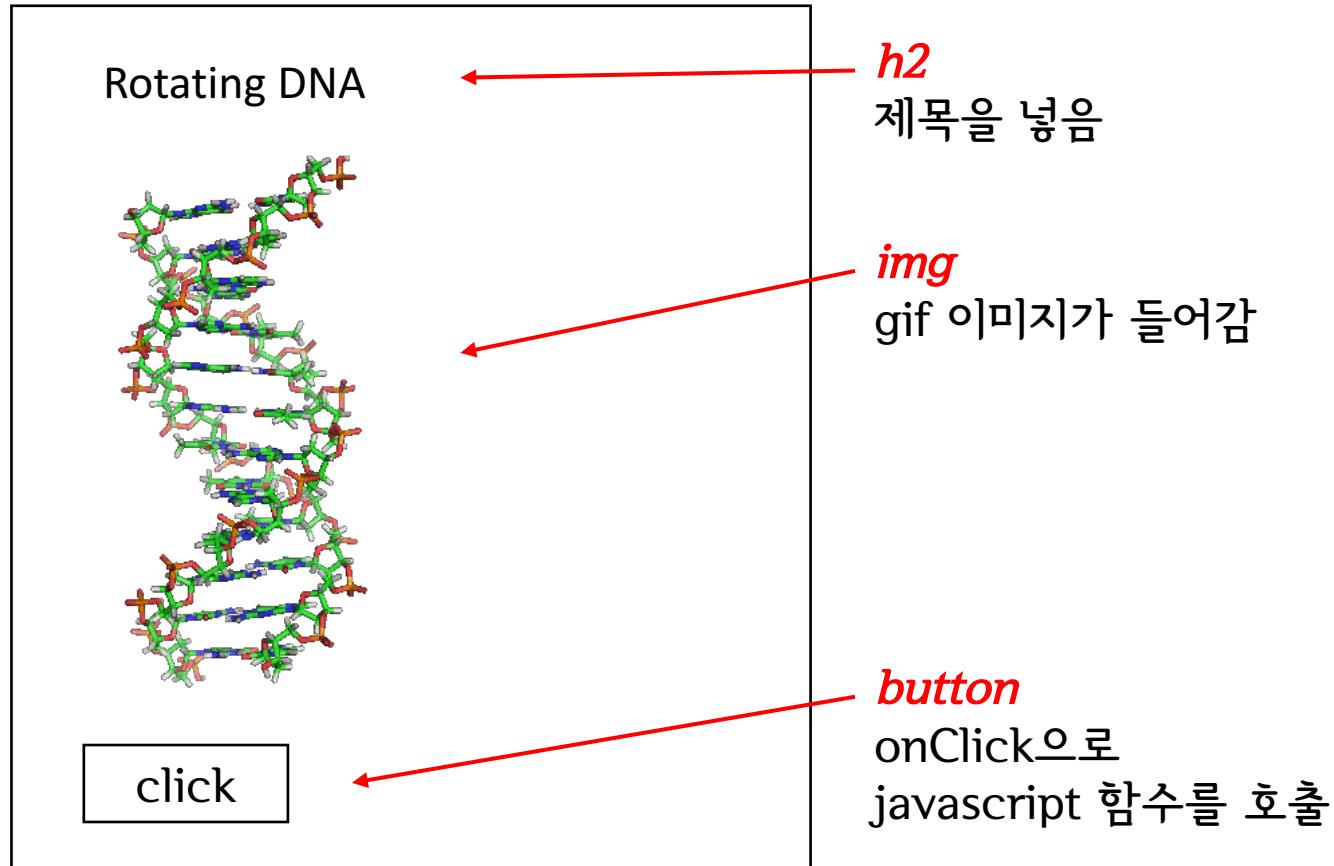
- JavaScript 함수

```
function triple(seq) {  
    for(i=0; i<seq.length; i+=3) {  
        console.log(seq.slice(i, i+3));  
    }  
}
```

```
> var seq = "ATGTTATAG";  
> triple(seq)  
ATG  
TTA  
TAG
```



JavaScript 간단 예제



한 쪽 방향으로 돌아가는 DNA gif 이미지인 dna_rotating_1.gif 가 있다.

이미지 아래의 click 버튼을 누르면 반대 방향으로 돌아가는 DNA gif 이미지인 dna_rotating_2.gif 로

다시 버튼을 누르면 dna_rotating_1.gif 로 바뀌는 페이지를 만들어보자.

이미지 출처

<http://bestanimations.com/Science/Biology/DNA/DNA.html>



JavaScript 간단 예제

rotating_dna.html — Bioinformatics_WebProgramming

rotating_dna.html x

html_css_js_bootstrap_jquery > rotating_dna.html > ...

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>돌아가는 DNA</title>
6      </head>
7      <body>
8          <h2>Rotating DNA</h2>
9          <!--src에 dna_rotate_1.gif를 넣습니다-->
10         <img id="dna_image" src="">
11         <br>
12         <!--button 속성으로 onclick="reverse()" 라고 써넣습니다.-->
13         <button id="dna_button" >click</button>
14
15     <script>
16         function reverse() {
17             var image_src = document.getElementById("dna_image").getAttribute('src');
18             var new_image_src = "";
19             console.log("button clicked: "+image_src);
20             if(image_src == "dna_rotate_1.gif") {
21                 new_image_src = "dna_rotate_2.gif";
22             } else {
23                 new_image_src = "dna_rotate_1.gif";
24             }
25             document.getElementById("dna_image").src=new_image_src;
26         }
27     </script>
28     </body>
29 </html>
30 |
```

x 0 ▲ 0 ↻ html | ↻ rotating_dna.html Ln 30, Col 1 Spaces: 4 UTF-8 LF HTML 😊 🔔 1

rotating_dna.html
파일을 수정하여 만들어보자

정답은
rotating_dna.html_ori
파일을 참조하자

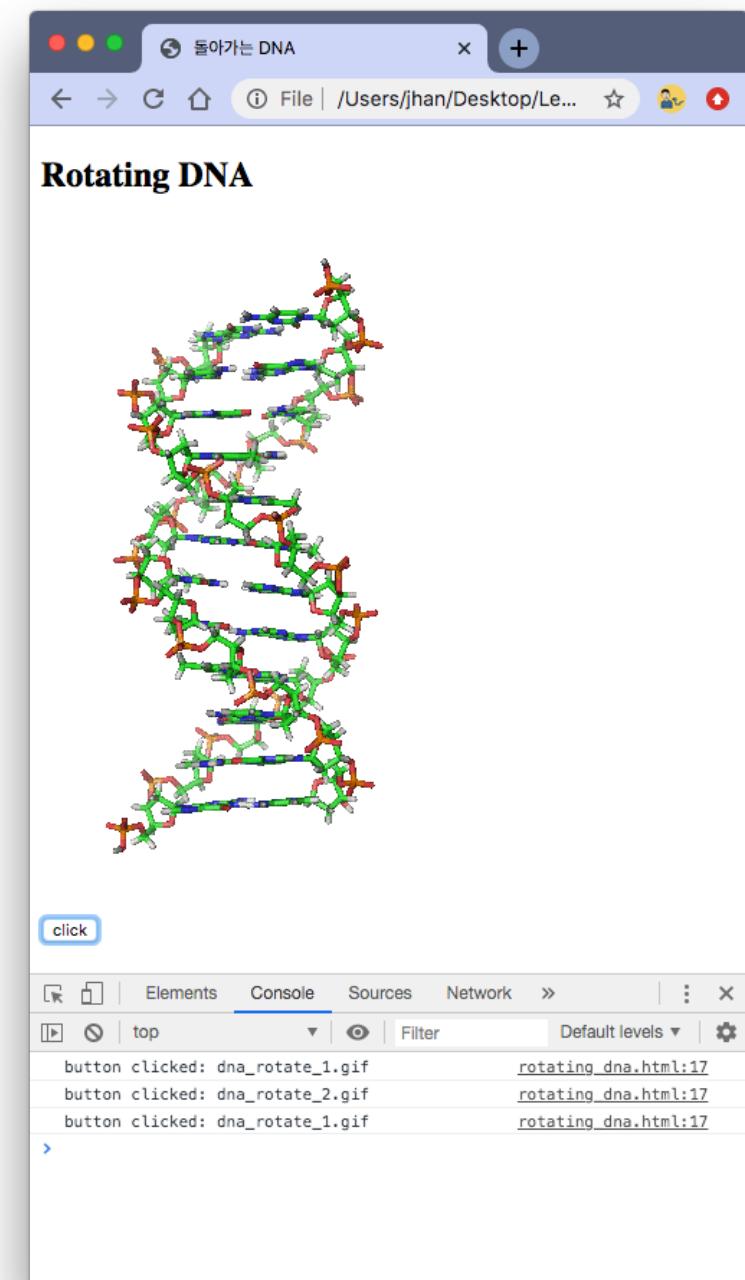


JavaScript 간단 예제

rotating_dna.html — Bioinformatics_WebProgramming

```
<rotating_dna.html>
01.html > <rotating_dna.html> ...
1   <!DOCTYPE html>
2   <html>
3       <head>
4           <meta charset="utf-8">
5           <title>돌아가는 DNA</title>
6       </head>
7       <body>
8           <h2>Rotating DNA</h2>
9           
10          <br>
11          <button id="dna_button" onclick="reverse()">click</button>
12
13         <script>
14             function reverse() {
15                 var image_src = document.getElementById("dna_image").getAttribute('src');
16                 var new_image_src = "";
17                 console.log("button clicked: "+image_src);
18                 if(image_src == "dna_rotate_1.gif") {
19                     new_image_src = "dna_rotate_2.gif";
20                 } else {
21                     new_image_src = "dna_rotate_1.gif";
22                 }
23                 document.getElementById("dna_image").src=new_image_src;
24             }
25         </script>
26     </body>
27 </html>
```

Ln 28, Col 1 Spaces: 4 UTF-8 LF HTML ⚡ 🔔



04 Bootstrap

부트스트랩은 트위터에서 개발한 UI (User Interface) 라이브러리

반응형 웹사이트 개발에 효과적 (데스크탑, 태블릿, 모바일)

그리드 시스템, 레이아웃, 버튼, 입력창 등을 제공

간단하면서도 멋진 페이지를 만들 수 있게 도와줌

참고 사이트

<https://www.w3schools.com/bootstrap/default.asp>

05 jQuery

JavaScript 라이브러리로 표준이 될 정도로 높은 사용률
매우매우매우 편함

```
// 표준 JavaScript
document.getElementById("btn").click(function(){ ... });
```

```
// jQuery
$("#btn").click(function(){ ... });
```

jQuery

<https://www.w3schools.com/jquery/default.asp>

jQuery 예제

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
      $(document).ready(function(){
        // DO SOMETHING
      });
    </script>
  </head>
  <body>
  </body>
</html>
```

jQuery button click

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
    $("btn").click(function(){
        // DO SOMETHING
    });
});
</script>
```

```
<head>
<meta charset="utf-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
var n = 0;
$(document).ready(function(){
    $("#btn").click(function(){
        console.log(n);
        if(n == 0){
            $("#protein").animate({"left":'+=300px'});
            n += 1;
        } else if(n == 1) {
            $("#protein").animate({"top":"+=300px"});
            n += 1;
        } else if(n == 2) {
            $("#protein").animate({"left":"-=300px"});
            n += 1;
        } else {
            $("#protein").animate({"top":"-=300px"});
            n = 0;
        }
    });
});
</script>
</head>
```

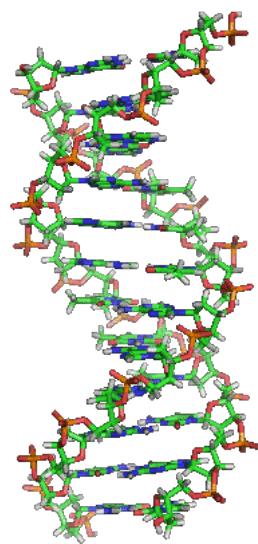
jquery_btn.html
에 다음 내용을
채워보자



```
<body>
    <button id="btn">GO</button>
    <div id="protein" style="height:100px;width:100px;position:absolute;">
        
    </div>
</body>
```

jQuery로 적용

Rotating DNA



click

이전에 만든 rotating_dna.html 파일을
rotating_dna_jquery.html로 복사 한 후
복사한 파일의 script 부분을 jquery로 바꾸어 보자

- 태그의 속성 가져오는 방법

```
$("#id").attr("attr_name");
```

- 태그의 속성을 지정하는 방법

```
$("#id").attr("attr_name", value);
```

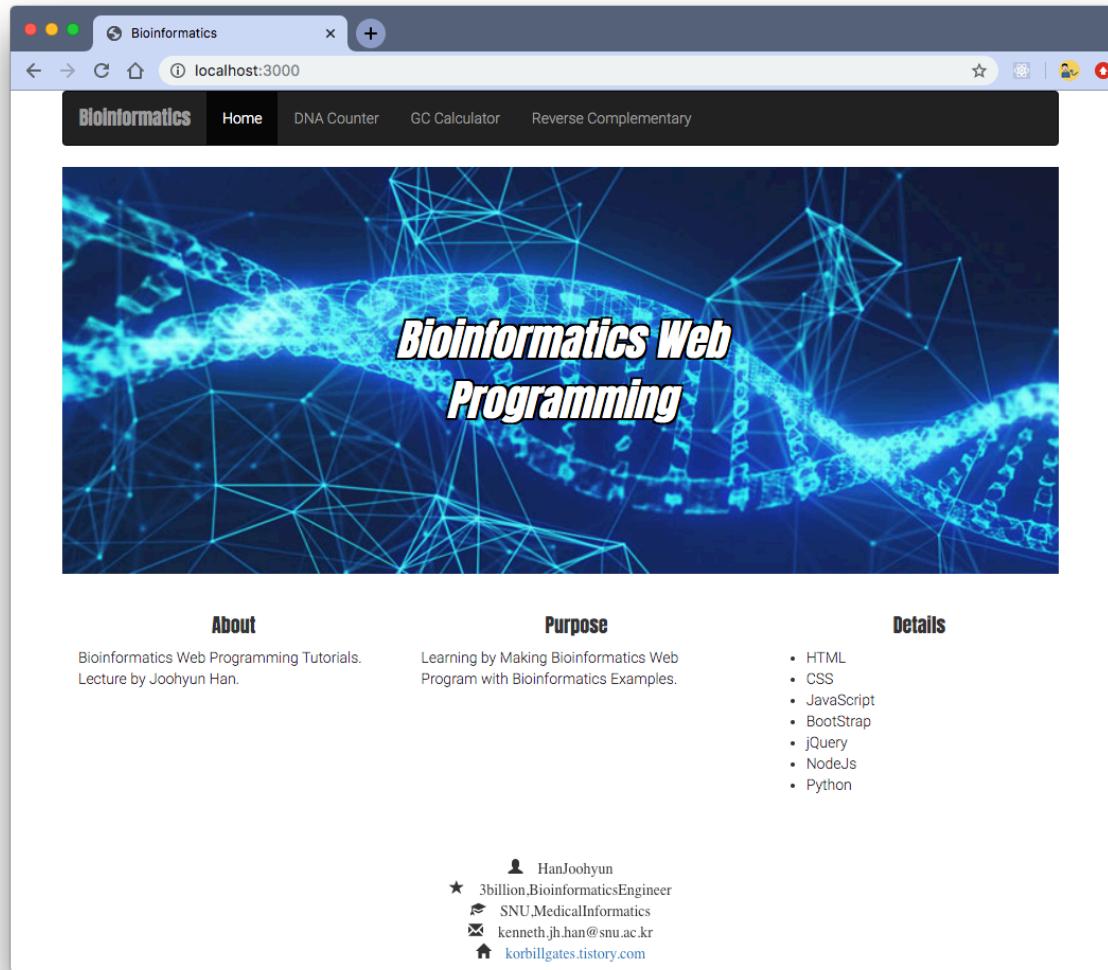
jQuery로 만들면 확실히 쉽다!

이미지 출처

<http://bestanimations.com/Science/Biology/DNA/DNA.html>

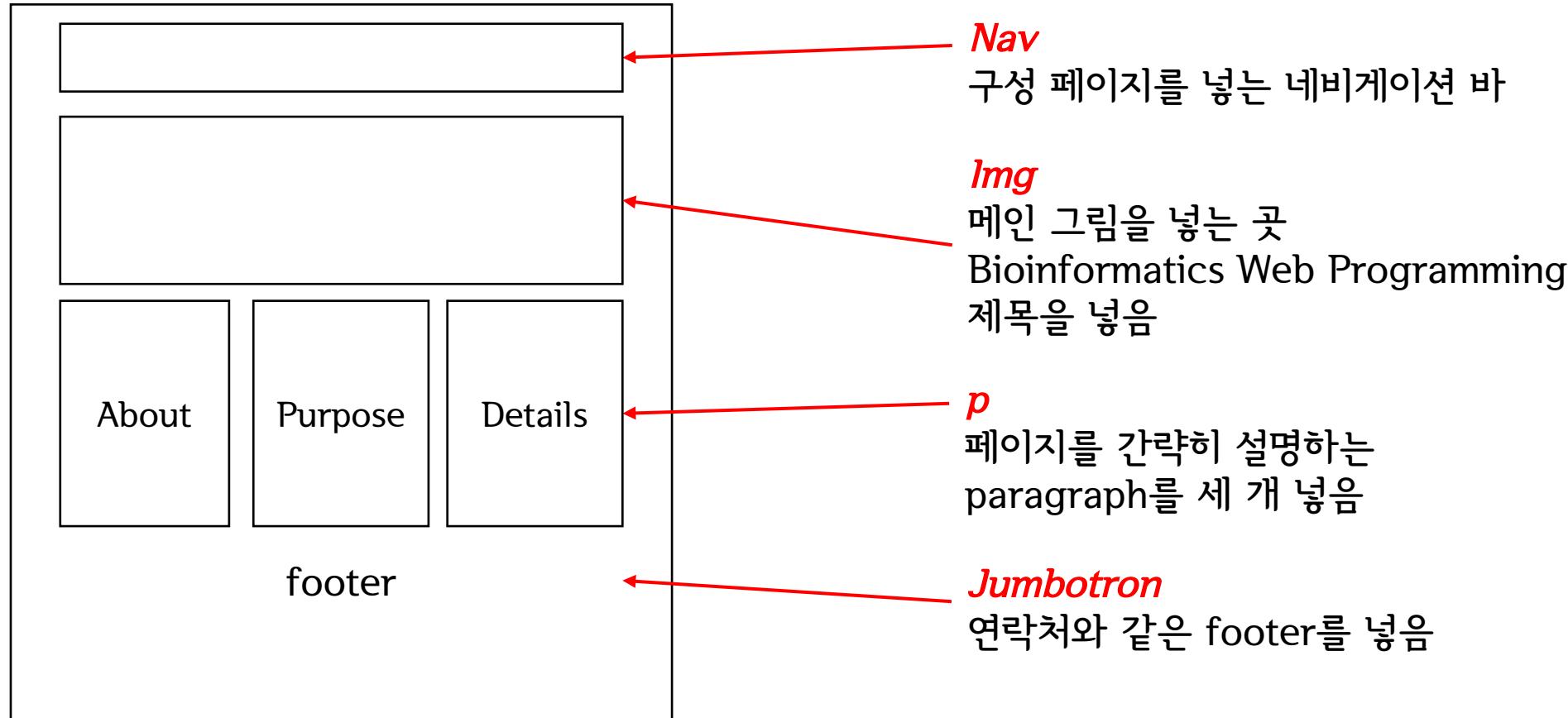


우리가 만든 목표

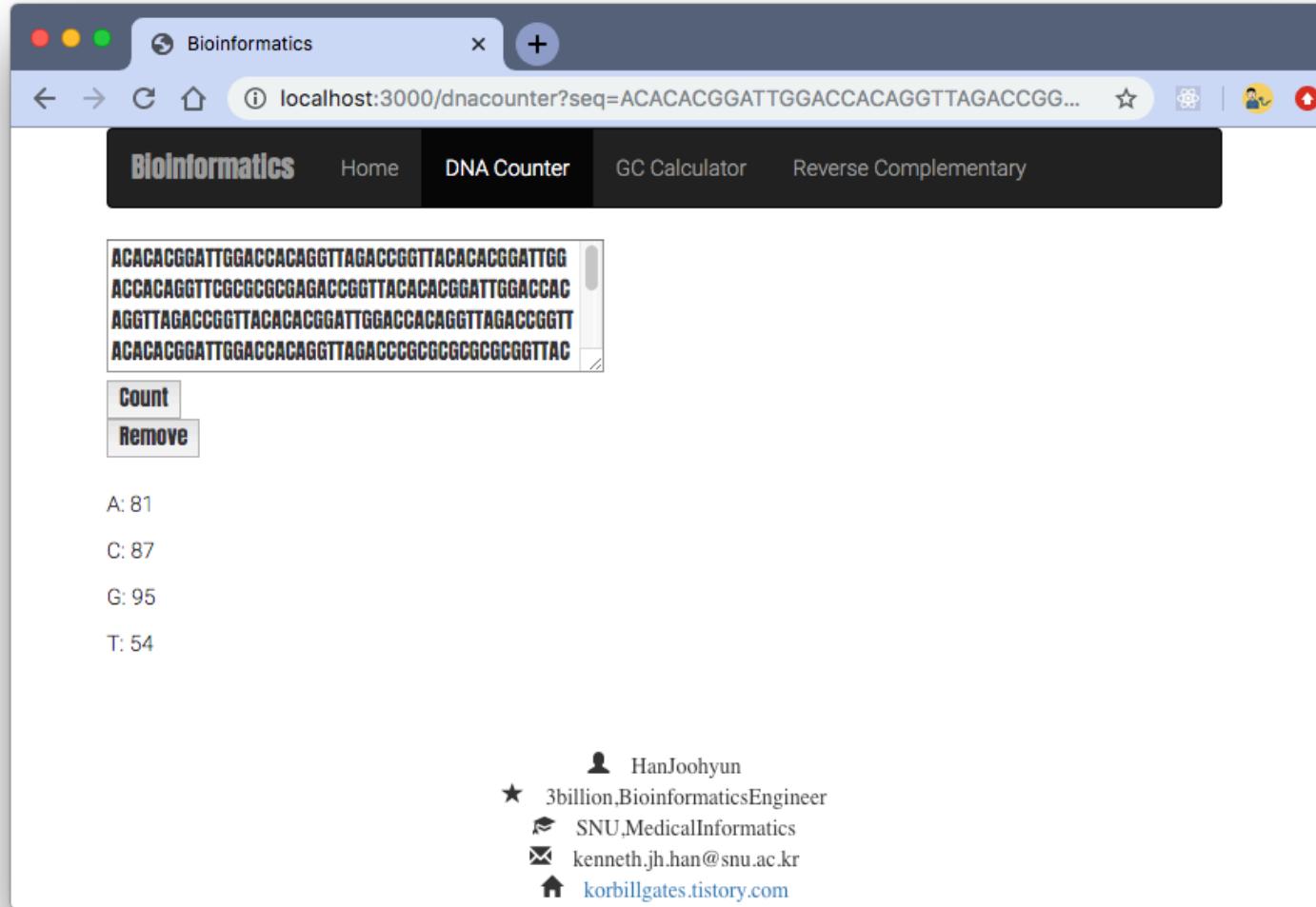


- 페이지 구성
- index
- dnaCounter
- gcCalculator
- reverseComplementary

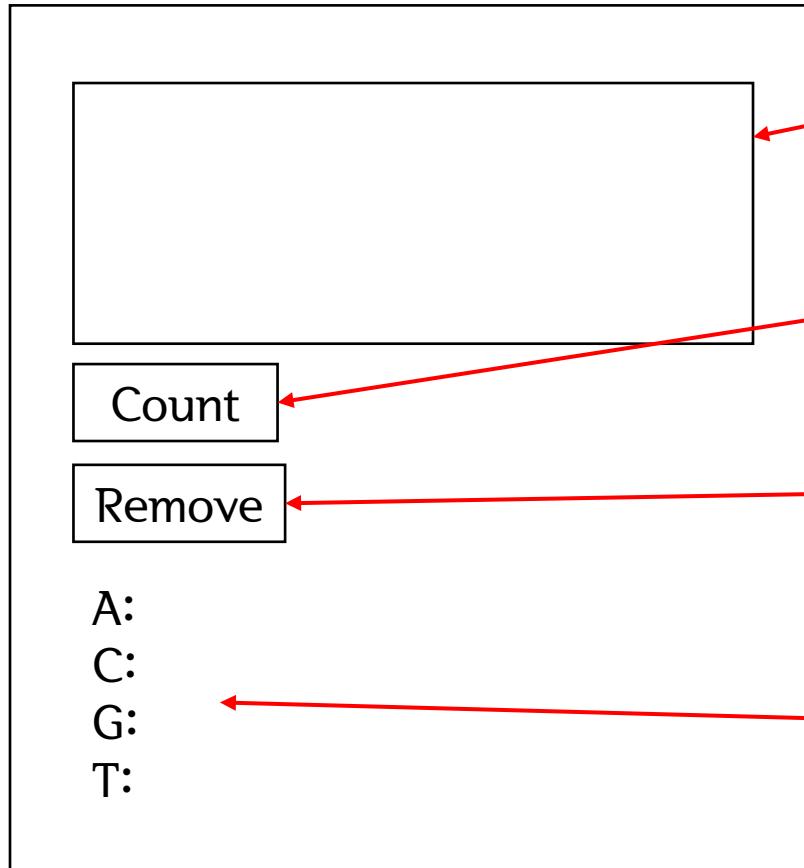
Index



우리가 만든 목표 - DNA counter



DNA counter



Text area

DNA 서열을 넣는 곳

Button

Count 버튼을 누르면 text area에서 A,C,G,T 를 세어 아래에 출력한다.

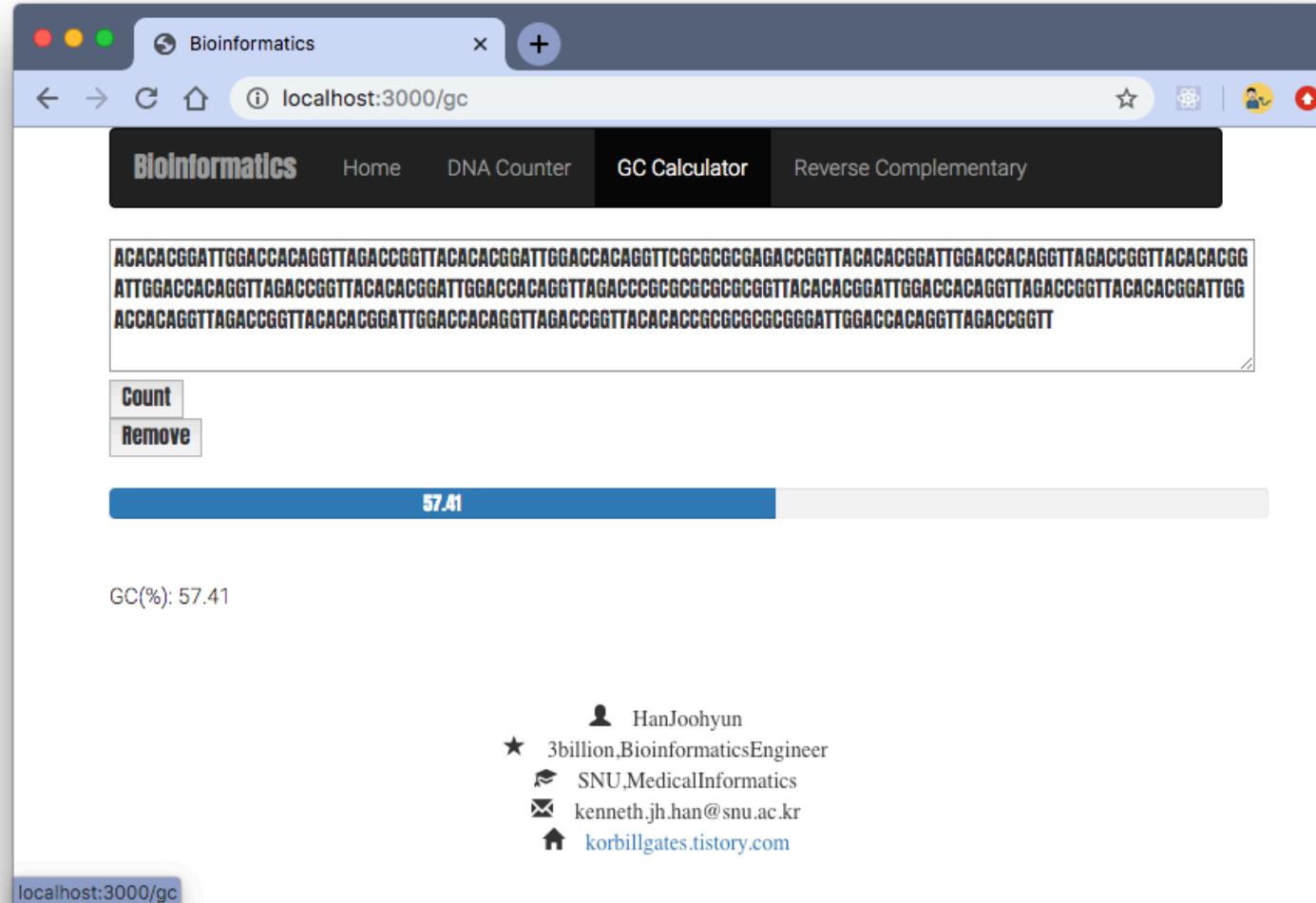
Button

Remove 버튼을 누르면 text area의 내용을 모두 삭제한다.

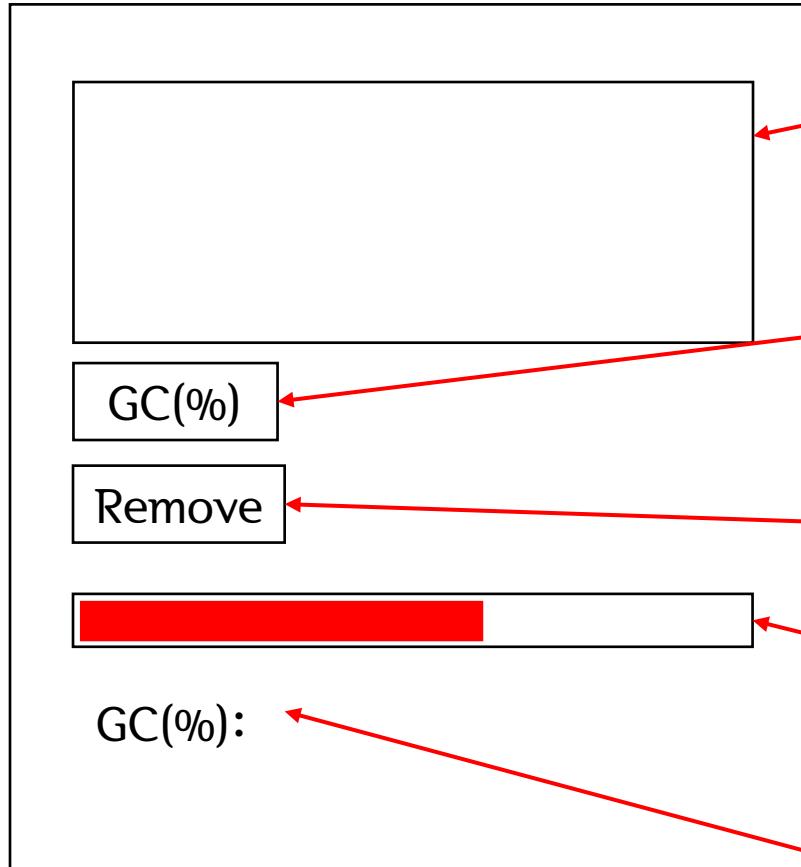
결과 내용이 표시되는 곳

미리 A: C: G: T: 를 써둔다.

우리가 만들 목표 - GC Calculator



GC(%) Calculator



Text area

DNA 서열을 넣는 곳

Button

Count 버튼을 누르면 text area에서 GC(%)를 계산하여 아래에 숫자로 출력 및 Progress bar에 표기한다.

Button

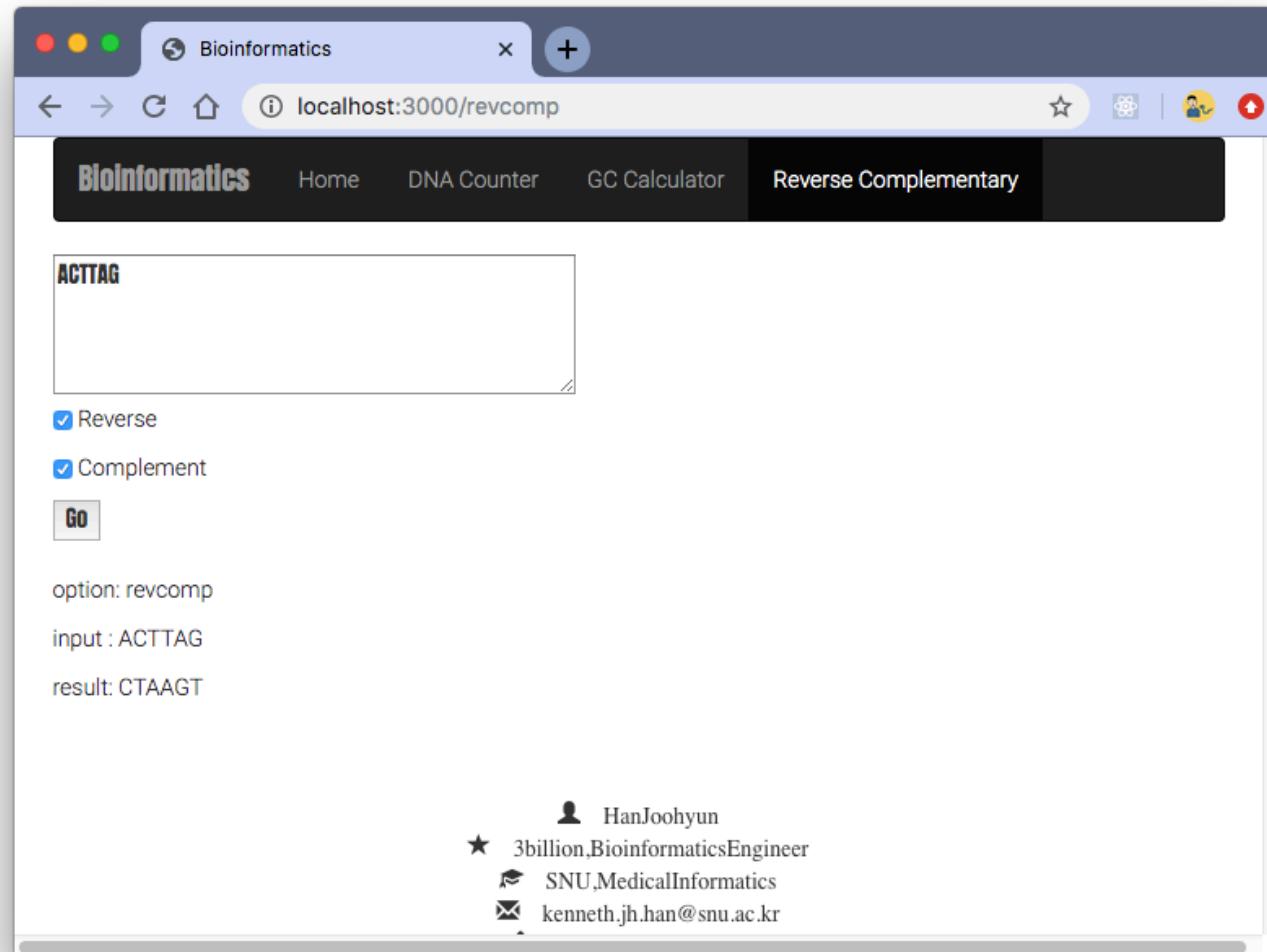
Remove 버튼을 누르면 text area의 내용을 모두 삭제한다.

Progress bar

결과 내용이 progress bar에 표시되는 곳

결과 내용이 표시되는 곳
소수점 둘째자리까지 표기한다.
미리 GC(%): 를 써둔다.

우리가 만든 목표 - Reverse Complement



Reverse Complementary Sequence

The diagram shows a user interface for generating reverse complementary DNA sequences. It consists of a large rectangular frame containing several input fields and controls. At the top left is a long horizontal input field. Below it is another long horizontal input field. To the left of these are two checkboxes: 'Reverse' and 'Complement'. At the bottom left is a rectangular button labeled 'Go'. Red arrows point from the right side of the image to various elements: one arrow points to the top input field with the label 'Input DNA 서열을 넣는 곳'; another arrow points to the second input field with the label 'Input 결과 서열이 나오는 곳'; a third arrow points to the 'Reverse' checkbox with the label 'Checkbox 어떤 기능을 구현할지 결정'; a fourth arrow points to the 'Complement' checkbox with the label 'Label 글씨는 label로 입력'; and a fifth arrow points to the 'Go' button with the label 'Button Go 버튼을 누르면 Checkbox를 확인하여 적절한 결과를 결과 서열이 나오는 곳에 넣는다'.

Input
DNA 서열을 넣는 곳

Input
결과 서열이 나오는 곳

Checkbox
어떤 기능을 구현할지 결정

Label
글씨는 label로 입력

Button
Go 버튼을 누르면 Checkbox를 확인하여
적절한 결과를 결과 서열이 나오는 곳에
넣는다
Checkbox가 모두 체크되어 있지 않다면
서열 그대로를 결과로 출력한다.

Node.js

2009년 5월 소개됨

JavaScript 엔진인 google V8에 비동기 이벤트 처리 라이브러리를 결합
웹 브라우저에서만 구동하던 JavaScript를 서버에서 실행할 수 있게 됨
리눅스, 맥, 윈도우 모두 지원

매우 간단하게 웹서버를 만들 수 있으면서도
웹 개발자라면 사용할 줄 아는 JavaScript로 구현되며
강력한 패키지 저장소인 npm의 활발한 업데이트

참고

<https://nodejs.org/en/>

Express.js

Node.js에서 가장 인기있는 웹 프레임워크

웹 프레임워크란 웹 서비스를 개발하기 위한 뼈대로
Java Spring, Python Django, PHP Laravel, Ruby Ruby on Rails
그리고 Node.js의 Express.js가 있다.

참고

<https://expressjs.com/>

nodejs, expressjs 설정

<https://nodejs.org/en/download/>

node js 다운로드 및 설치

- 설치 확인 방법

```
$ node
```

```
> console.log("hello")
```

```
hello
```

- expressjs 설치

```
$ npm install -g express-generator
```

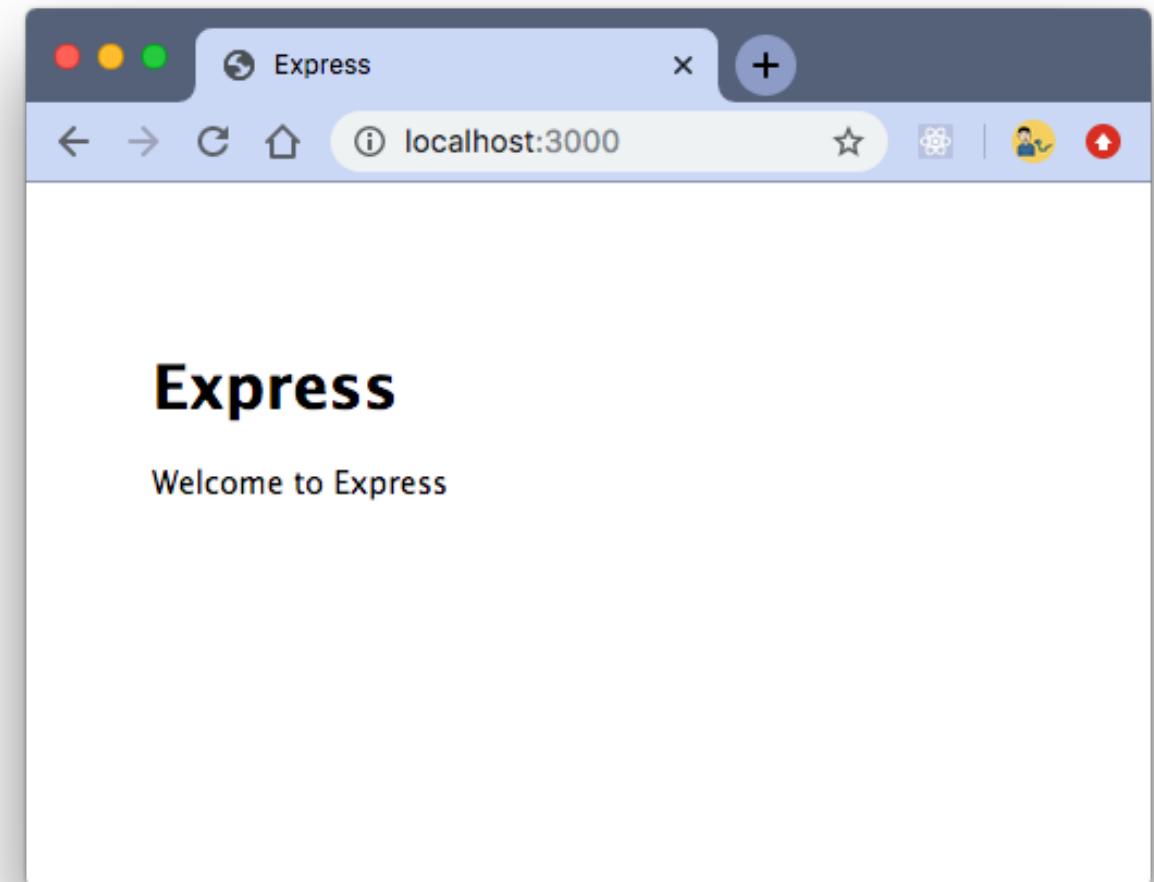
start from scratch

```
$ express --view=ejs bioinformatics  
(엔터 .. 엔터)
```

```
$ cd bioinformatics
```

```
$ npm install
```

```
$ npm start
```



완성될 디렉토리 구조와 파일들 살펴보기

```
. └── routes (app.js에서 설정한 접속 라우팅)
      ├── dnacounter.js
      ├── gc.js
      ├── getpage.js
      ├── index.js
      ├── revcomp.js
      └── users.js
  └── views (app.js에서 설정한 렌더링 뷰 페이지들)
      ├── dnacounter.ejs
      ├── error.ejs
      ├── gc.ejs
      ├── index.ejs
      └── revcomp.ejs
  ├── app.js (전체 구조 정보)
  ├── bin
  |   └── www (접속 정보 등)
  ├── node_modules (설치한 노드 모듈)
  ├── package-lock.json
  └── package.json
  └── public (static 파일들이 저장될 장소)
      ├── images
      |   └── back.png (배경 이미지)
      ├── javascripts
      ├── scripts
      |   └── revComp.py (역상보 서열을 만드는 스크립트)
      └── stylesheets
          └── style.css (스타일 css)
```

app.js

```
var index = require('./routes/index');
var dnacounter = require('./routes/dnacounter');
var gc = require('./routes/gc');
var revcomp = require('./routes/revcomp');

app.use('/', index);
app.use('/dnacounter', dnacounter);
app.use('/gc', gc);
app.use('/revcomp', revcomp);
```

빨강이를 추가한다



app.js

```
var index = require('./routes/index');
//var dnacounter = require('./routes/dnacounter');
//var gc = require('./routes/gc');
//var revcomp = require('./routes/revcomp');
```

```
app.use('/', index);
//app.use('/dnacounter', dnacounter);
//app.use('/gc', gc);
//app.use('/revcomp', revcomp);
```

그리고 우선 빨강이를
// 주석 처리하자
하나씩 구현하면서
주석 처리를 지울거다~



routes/index.js

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```

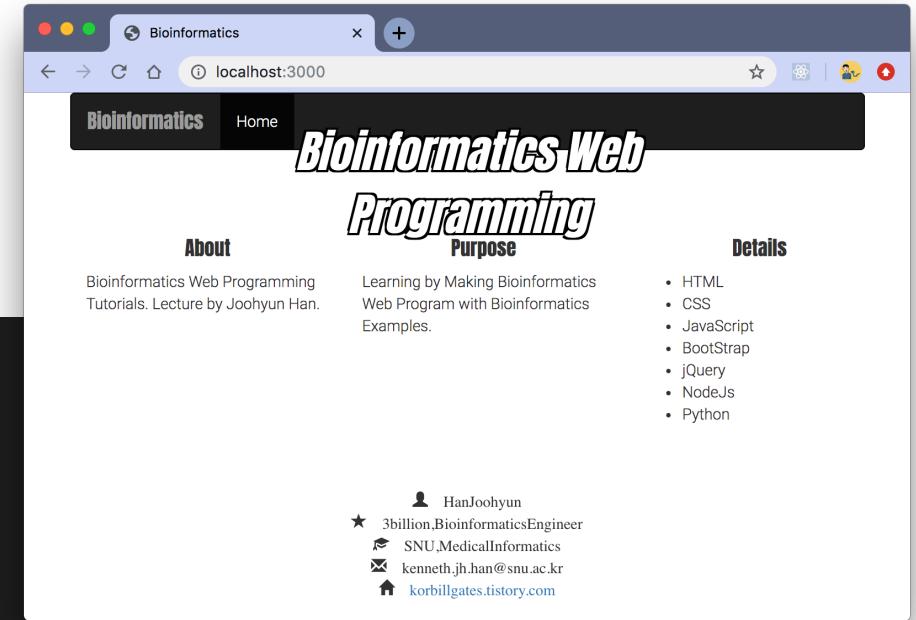
app.js에서
app.use('/', index);
가 있는데 주소의 루트 (/)
로 접근하면
index.js로 가라는 얘기다

index.js에서는
/로 받는 경우
res.render('index', {})
가 있는데
view의 index를
찾아가라는 얘기다



views/index.ejs

```
<body>
  <div class="container">
    <nav class="navbar navbar-inverse">
      <div class="container-fluid">
        <div class="navbar-header">
          <a class="navbar-brand" href="/">Bioinformatics</a>
        </div>
        <ul class="nav navbar-nav">
          <li class="active"><a href="/">Home</a></li>
          <!--li a /dnacounter DNA Counter-->
          <!--li a /gc GC Calculator -->
          <!--li a /revcomp Reverse Complement -->
        </ul>
      </div>
    </nav>
    <div class="cont">
      <img width="100%" style="opacity:0.95;" src=""> <!--images/back.png 을 넣어줌-->
      <div class="centered outline" style="font-size:40px;"><i>Bioinformatics Web Pro</i>
    </div>
```



주변을 참고하여
body에서 주석으로
처리된 미완성 부분을
채워보자



routes/dnacounter.js

js dnacounter.js ×

test ▶ routes ▶ js dnacounter.js ▶ ...

```
1  var express = require('express');
2  var router = express.Router();
3
4  router.get('/', function(req, res, next) {
5      console.log("## get");
6      // SET "a, c, g, t" to 0
7      var /**/;
8      // SET seq to req.query.seq
9      var seq = /**/;
10     // seq.toUpperCase()
11     seq = /**/;
12     console.log("seq: "+seq);
13     // count A, C, G, T
```

```
14     if(seq){
15         for(*; ;*) {
16             if(seq[i] == "A"){
17                 a += 1;
18             } else if(***) {
19                 /**;
20             } else if(***) {
21                 /**;
22             } else if(***) {
23                 /**;
24         }
25     }
26 }
27
28     // render to dnacounter
29     // pass value. cntA: a, cntC: c, ...
30     res.render('dnacounter', { /* */ });
31 });
32
33 module.exports = router;
34
```

dna counter
부터
만들어보자



views/dnacounter.ejs

```
<div class="container">
  <div class="row">
    <!-- form 을 만든다 action="/dnacounter" method="get"-->
    <!-- textarea 의 id="seq" name="seq" rows="" 와 cols="" 는 적당한 숫자로 만들어본다 -->
    <!-- button 의 id="countBtn" 으로 만들어본다-->
    <!-- form 을 닫는다 -->
    <!-- button 의 id="removeBtn" 으로 만들어본다 -->
  </div>
  <br>
  <div class="row">
    <p>A: <%= cntA %></p>
    <!-- C, G, T 도 참고하여 만든다 -->
  </div>
</div>
```

```
<script>
  // jQuery로 #removeBtn이 클릭될 때 #seq의 val을 "" 로 설정되게 한다.
</script>
```

DNA counter

The screenshot shows a web browser window titled "Bioinformatics" at the URL "localhost:3000/dnacounter?seq=ACACACGGATTGGACCACAGGTTAGACCGG...". The browser has a dark theme. The main content area contains a DNA sequence in a code editor-like interface:

```
ACACACGGATTGGACCACAGGTTAGACCGGTTACACACGGATTGG  
ACCCACAGGTTCGCGCGCGAGACCGGTTACACACGGATTGGACCAC  
AGGTTAGACCGGTTACACACGGATTGGACCACAGGTTAGACCGGTT  
ACACACGGATTGGACCACAGGTTAGACCCCGCGCGCGCGCGCGGTAC
```

Below the sequence are two buttons: "Count" and "Remove". Underneath the buttons, the base counts are displayed:

- A: 81
- C: 87
- G: 95
- T: 54

At the bottom, there is a footer with the author's information:

HanJooHyun
★ 3billion,BioinformaticsEngineer
 SNU,MedicalInformatics
 kenneth.jh.han@snu.ac.kr
 korbillgates.tistory.com

GET 을 사용했다는
것에 주목!



routes/gc.js

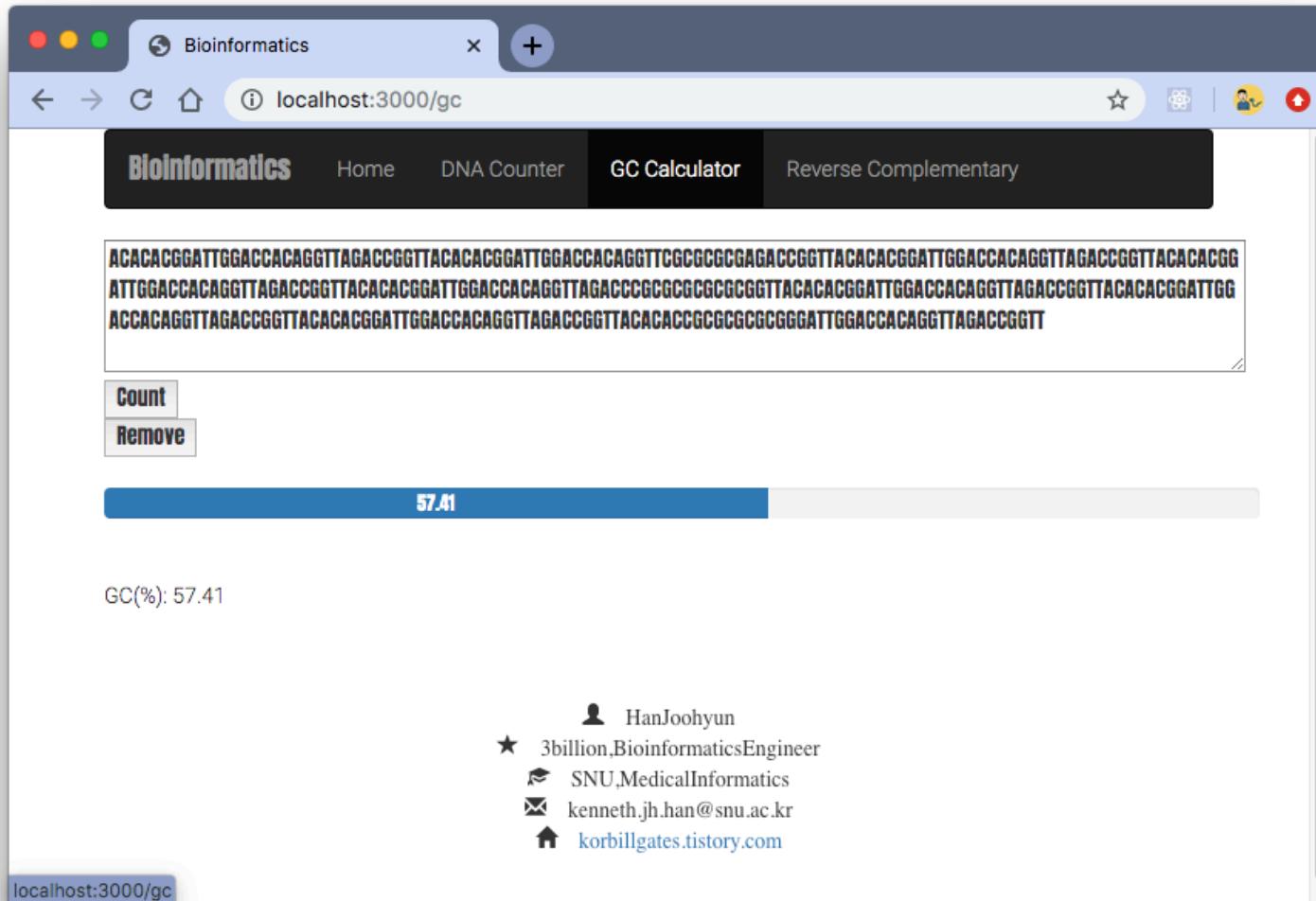
```
JS gc.js      x  
test ▶ routes ▶ JS gc.js ▶ ⚏ router('') callback  
1  var express = require('express');  
2  var router = express.Router();  
3  
4  // get  
5  router.get('/', function(req, res, next) {  
6    res.render('gc', {gcp:0});  
7  });  
8  
9  // post  
10 router./**/('/' , function(req, res, next) {  
11   console.log("## post");  
12   var a=0, c=0, g=0, t=0, gcp=0;  
13   /* seq set to req.body.seq */  
14   var seq = /**/;  
15   seq = seq.toUpperCase();  
16   console.log(req.body);  
17   console.log("seq: "+seq);  
18   if(seq){  
19     for(i=0;i<seq.length;i++) {  
20       if(seq[i] == "A"){  
21         a += 1;  
22       } else if(seq[i] == "C") {  
23         c += 1;  
24       } else if(seq[i] == "G") {  
25         g += 1;  
26       } else if(seq[i] == "T") {  
27         t += 1;  
28       }  
29     }  
30     /* calc gc percent */  
31     gcp = /**/;  
32     /* round using Math.round() */  
33     gcp = /**/;  
34     console.log("a: "+a+",c: "+c+",g: "+g+",t: "+t);  
35     console.log('gc: '+gcp);  
36   }  
37   /* render gcp : gcp */  
38   res.render('gc', {/**/});  
39 };  
40  
41 module.exports = router;
```

views/gc.ejs

```
<div class="row">
  <div class="progress">
    <!-- bootstrap의 progress-bar 페이지를 참조하여 class에 적당한 값을 넣어주세요-->
    <div id="progress-bar" class="" role="progressbar"
      aria-valuenow="<%=gcp%>" aria-valuemin="0" aria-valuemax="100"
      style="width:<%=gcp%>%
        <%=gcp%>
    </div>
  </div>
</div>
<br>
<div class="row">
  <!-- 서버에서 넘어오는 gc percent 값을 GC(%): 다음에 표시해주세요 -->
  <p>GC(%): </p>
</div>
</div>
```

https://www.w3schools.com/bootstrap/bootstrap_progressbars.asp

GC Calculator



GET과 POST의 차이점에 주목!



실망한 뱀뱀이

실망한 뱀뱀이를 달래줘야 하는데
JavaScript 뿐인 환경에서
Python 을 어떻게 접목시킬 수 있을까요?

서버가 있다는 점의 진가가
여기서 발휘됩니다!

파이썬을 사용하여
reverse complement 서열을 구하고
결과를 express로 넘겨서
페이지를 완성해봅시다!

웹 브라우저와 서버간의 상호작용을 확인해보세요!

ㅠㅠ
파이썬은 안쓰는건가?



routes/revcomp.js

```
JS revcomp.js ×

test › routes › JS revcomp.js › ...
  1  var express = require('express');
  2  var router = express.Router();
  3  var exec = require("child_process").exec;
  4
  5  /* GET home page. */
  6  router.get('/', function(req, res, next) {
  7    console.log("## GET: revcomp");
  8    res.render('revcomp', { input: "", result: "", option: "" });
  9  });
 10
 11 router.post('/', function(req, res, next) {
 12   console.log("## POST: revcomp");
 13   var result = "";
 14   var seq = req.body.seq;
 15   var option = "";
 16   seq = seq.toUpperCase();
 17   if(req.body.rev !== undefined)
 18     option += /**/; // add option "rev"
 19   if(req.body.comp !== undefined)
 20     option += /**/; // add option "comp"
 21
```

```
22  /* set cmd as python public/script/revComp.py [seq] [option] */
 23  var cmd = /**/;
 24  console.log("cmd: "+cmd);
 25  console.log("seq: "+seq);
 26  console.log("opt: "+option);
 27  if(seq){
 28    exec(cmd, function(err,stdout,stderr){
 29      console.log("stdout: "+stdout);
 30      console.log("stderr: "+stderr);
 31      res.render('revcomp', {
 32        input: seq, result: stdout, option: option
 33      });
 34    });
 35  }
 36
 37  //res.render('revcomp', { result: result });
 38  });
 39
 40 module.exports = router;
```

public/srcipts/revComp.py

```
1 #!/usr/bin/python
2
3 import sys
4
5 if len(sys.argv) != 3:
6     sys.exit()
7
8 def rev(seq):
9     revSeq = ""
10    ## seq의 순서를 뒤집은 서열인 revSeq을
11    ## 반환하는 함수를 만들어보세요
12    return revSeq
13
14 def comp(seq):
15    compSeq = ""
16    ## seq의 상보적 서열인 compSeq을
17    ## 반환하는 함수를 만들어보세요
18    return compSeq
```

```
19
20 seq = sys.argv[1]
21 mode = sys.argv[2]
22 result = ""
23
24 if mode == "rev":
25     result = rev(seq)
26 elif mode == "comp":
27     result = comp(seq)
28 elif mode == "revcomp":
29     result = comp(rev(seq))
30
31 print(result)
```

결과는 print로 출력하여
nodejs가 stdout으로
받을거다



public/revcomp.ejs

```
<div class="row">
  <!-- form 의 메서드는 post 로 합니다 -->
  <form action="/revcomp" method="">
    <textarea id="seq" name="seq" rows="4" cols="50"></textarea><br>
    <!-- 다음 p 태그에 두 개의 체크박스를 만들어 보세요 -->
    <!-- input type="checkbox" name="rev" value="" -->
    <!-- input type="checkbox" name="comp" value="" -->
    <!-- button id="countBtn" 인 버튼을 만들어보세요 -->
  </form>
</div>
```

체크박스 참고

https://www.w3schools.com/bootstrap/bootstrap_forms_inputs.asp

Reverse Complement

Screenshot of a web browser showing a Bioinformatics tool at localhost:3000/revcomp. The input sequence is ACTTAG. The 'Reverse' and 'Complement' checkboxes are checked. The result is CTAAGT.

Bioinformatics Home DNA Counter GC Calculator Reverse Complementary

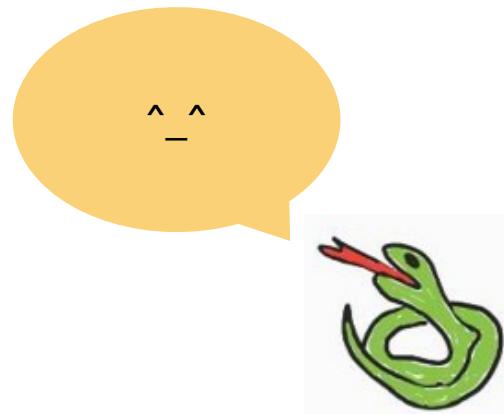
ACTTAG

Reverse
 Complement

Go

option: revcomp
input : ACTTAG
result: CTAAGT

HanJoohyun
★ 3billion,BioinformaticsEngineer
✉ SNU,MedicalInformatics
✉ kenneth.jh.han@snu.ac.kr



강사 소개	
이름	한주현
소속	전) 마크로젠 데이터분석부 현) 쓰리빌리언 Bioinformatics Engineer 현) 서울대학교 의료정보학 전공 이규언 교수님 연구실
메일	kenneth.jh.han@snu.ac.kr
웹페이지	https://korbillgates.tistory.com



짧은 시간이기에 못다한 얘기들이 많은데
여러분들과 헤어지는게 아쉽네요
필요한 일 있으시면 언제든지 연락주세요 😊

그럼 나중에 또 만나요 !!



더 읽어보기

웹은 인터랙티브한 프론트 영역과 논리를 뒷받침하는 서버 영역이 있습니다.

개발하는 내용을 정확히 알고 제대로 만드는 능력과

최신 기술에 대한 관심과 이를 익혀가는 능력이

제 개인적으로 웹 개발자의 역량 중 하나라 생각합니다.

기본적인 HTML/CSS/Javascript

프론트엔드, 백엔드에 대한 개념

GET/POST 요청의 차이점

Model View Control에 대한 개념

간단한 jQuery 사용

AngularJS, ReactJS, VueJS 와 같이 요새 핫한 프론트엔드 라이브러리

Amazon AWS (EC2, DynamoDB, S3) 에 대한 사용방법

등의 키워드를 시작으로 학습 하시는 것을 추천드립니다.

화이팅!

