

Part3 :

- POST 방식을 사용
- 사용자가 제공하는 값을 기반으로, 다이내믹 HTML 콘텐츠를 생성
- 1. 사용자는 HTML양식에 접근할 수 있어야 한다. (part2에서 static 폴더에 저장한것)
 - 양식예시는 pdf 파일 3페이지 하단 참조.
 - 양식을 만드는 자세한 튜토리얼은 해당 링크 참조
 - https://www.w3schools.com/html/html_forms.asp
- 2. 양식은 post 방식을 이용해 사용자가 제공한 정보를 서버로 보낸다 (post method)
- 3. Action attribute 는 양식이 등록되었을 시, 양식데이터를 어디로 보낼지 결정한다.
- 4. 양식은 여러 필드들로 구성되어 있고, 이 필드들은 이름과 값을 정한다.
- 5. Form 을 Encode 를 하는 방법중 하나는 application/x-www-form-urlencoded.를 사용하는 것이다.
(4 페이지 참조)

예를들어, 사용자가 이름을 John 성을 Doe, 성별을 male라고 양식에 입력했을 시, 브라우저는 다음과 같은 요청을 생성한다. (4페이지 중단 참조)

```
POST /r1.html HTTP/1.1
Host: localhost:8001
Connection: keep-alive
Content-Length: 32
Cache-Control: max-age=0
Origin: http://localhost:8001
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4)
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://localhost:8001/f1.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8
Cookie: Webstorm-ea36d52b=7cf702c4-f920-40c7-acfd-a25b1d77b36c
fname=John&lname=Doe&gender=male
```

* part1에서는 GET 방식을 다뤘다면, 이번에는 POST방식을 다뤄야 한다. HTTP POST의 길이는 Content-Lengh에서 제공한다. Content-Type은 application/x-www-form-urlencoded. 으로 설정되어있어야한다.

Part3-2: 다이내믹 웹 서버는 HTTP POST request 를 처리해야한다. Request 의 유효성을 확인하고 , HTTP request Body 에 있는 name-value 짝을 딕셔너리(dictionary)에 파스(parse) 해야한다. 추가로 반드시 HTTP GET 방식과 POST 방식을 둘다 처리할 수 있어야한다.

- HTTP request 와 respond 를 다룰 딕셔너리를 만들었다면, 간단한 템플레이팅 언어 (https://wiki.python.org/moin/Templating#HTML_Shorthand_Processors)를 사용해 다이내믹 HTML contents 를 생성해야한다. (5 페이지 참조)
- 끝으로 HTML 페이지를 생성한다. (5 페이지 하단에 도움이 될만한 링크들이 있습니다.)