

Environment

window 10, Dev-C++

1. code and explanation(주석처리)

1)svg.h

```

1  #ifndef SVG_H
2  #define SVG_H
3
4  class SVG{
5  public:
6      vector<SVG*> svg_child; //svg자식을 저장할 수 있는 벡터
7      SVG(string tag_name_);
8      void set_if_selected(int if_selected_);
9      int get_if_selected();
10     int find_if_selected();
11     string get_tag_name();
12     void append(string tag_name_);
13     void select(string tag_name_);
14     void remove();
15     void end();
16     void enter(map<string, vector<string>>, string tag_name_, int bounded_csv_index_, string unique_id_);
17     void update(map<string, vector<string>>, string tag_name_, int bounded_csv_index_, string unique_id_);
18     void exit(map<string, vector<string>>, string tag_name_, string unique_id_);
19     void attr(string attr_name_, string attr_value_);
20     void tattr(string x_multiplier_, string y_multiplier_);
21     void dattr(vector<map<string, vector<string>>>, vector<string> field, string svg_attr_name_, string datum_field_name_);
22     void dattr(vector<map<string, vector<string>>>, vector<string> field, string svg_attr_name_, string datum_field_name_, float mul_, float add_);
23     void print(FILE *html_out_);
24     //void print();
25 private:
26     string tag_name_; //tag_name를 저장
27     int if_selected; //select가 된 상태라면 1, 자식svg가 select된 상태라면 2
28     int bounded_csv_index; //bounded csv의 index
29     string Unique_Id[3]; //Unique_Id[0]에는 CSV Unique id의 field name, Unique_Id[1]에는 CSV Unique id의 값
30     map<string, string> attribute; //svg의 attribute를 저장하는 map
31     map<string, string>::iterator attribute_iter;
32 };
33 #endif

```

2)svg.cpp

```

1  #include <iostream>
2  #include <string>
3  #include <vector>
4  #include <map>
5  #include <sstream>
6  using namespace std;
7
8  #include "SVG.h"
9
10 SVG::SVG(string tag_name_){
11     tag_name_ = tag_name_; //svg객체 생성시 tag name도 함께 저장
12     if_selected = 1; //svg객체 생성시 이 객체를 select함
13 }
14
15 void SVG::set_if_selected(int if_selected_){ //if_selected의 값 set
16     if_selected = if_selected_;
17 }
18
19 int SVG::get_if_selected(){ //if_selected의 값 get
20     return if_selected;
21 }
22
23 string SVG::get_tag_name(){ //tag_name의 값 get
24     return tag_name_;
25 }
26
27 void SVG::append(string tag_name_){ //append 함수
28     if(if_selected == 1){ //만약 select된 상태라면
29         svg_child.push_back(new SVG(tag_name_)); //svg자식을 추가
30         if_selected = 2; //그 후 자신의 if_selected는 2
31         return ;
32     }
33     else if(if_selected == 2){ //만약 자식중 선택된 것이 있다면
34         for(int i=0; i<svg_child.size(); i++){
35             svg_child[i]->append(tag_name_); //자식에 대해 append 실행
36         }
37     }
38     else{}
39 }
40

```

```

41 void SVG::select(string tag_name){ //select와 selectAll을 위한 함수
42     if(if_selected == 1){ //select된 상태라면
43         for(int j=0; j<svg_child.size(); j++){
44             if(svg_child[j]->get_tag_name() == tag_name){
45                 svg_child[j]->set_if_selected(1); //tag_name을 가진 자식을 selection으로 변경
46             }
47         }
48         if_selected = 2; //자신의 if_selected는 2로 변경
49         return ;
50     }
51     else if(if_selected == 2){ //자식이 선택된 상태라면
52         int check = 0;
53         for(int i=0; i<svg_child.size(); i++){
54             if(svg_child[i]->get_if_selected() != 0){
55                 check = 1; //자식이 선택되었다면 (1 or 2)
56                 svg_child[i]->select(tag_name); //자식에 대해 select할수 실행
57             }
58             if(check == 0){
59                 set_if_selected(2); //자식 중 select된 것이 없다면 자신의 if_selected는 2로 해서 selection scope는 자식으로 바꿈.
60                 return;
61             }
62         }
63     }
64     else{}
65 }
66

```

그림 3 select

```

67 void SVG::remove(){
68     if((if_selected == 2) && (svg_child.size() == 0)){ //만약 selection은 2인데 자식이 없다면
69         if_selected=1; //자신을 select하고 끝냄
70         return;
71     }
72     int index=0;
73     for(int j=0; j<svg_child.size(); j++){
74         if(svg_child[j]->get_if_selected() == 1){ //자식 중 if_selected가 1인 것이 있을 경우
75             index=1;
76             break;
77         }else if(svg_child[j]->get_if_selected() == 2){ //자식 중 if_selected가 2인 것이 있을 경우
78             index=2;
79             break;
80         }
81     }
82     if(index == 1 || index == 0){ //index가 0인 경우는 자식의 if_selected가 모두 0인 경우 (scope는 자식)
83         for(int j=0; j<svg_child.size(); j++){
84             if(svg_child[j]->get_if_selected()==1){
85                 svg_child.erase(svg_child.begin()+j); //selection이 1인 객체 모두 제거
86                 j--;
87             }
88         }
89         if_selected = 1; //자식이 제거되고 그 부모인 자신이 선택
90         return ;
91     }else if(index == 2){
92         for(int j=0; j<svg_child.size(); j++){
93             if(svg_child[j]->get_if_selected()==2){
94                 svg_child[j]->remove(); //자식 중 if_selected가 2이면 그 객체에 대해 remove할수 실행
95                 break;
96             }
97         }
98     }
99 }

```

그림 4 remove

```

101 void SVG::end(){ //remove와 유사
102     if((if_selected==2) && (svg_child.size() == 0)){
103         if_selected=1;
104         return;
105     }
106     int index=0;
107     for(int j=0; j<svg_child.size(); j++){
108         if(svg_child[j]->get_if_selected()==1){
109             index=1;
110         }
111     }
112     if(index == 1){
113         for(int j=0; j<svg_child.size(); j++){
114             if(svg_child[j]->get_if_selected()==1){
115                 svg_child[j]->set_if_selected(0);
116             }
117         }
118         if_selected = 1;
119         return ;
120     }
121     }else if(index == 0){
122         int no_selected_child = 1;
123         for(int j=0; j<svg_child.size(); j++){
124             if(svg_child[j]->get_if_selected()==2){
125                 svg_child[j]->end();
126                 no_selected_child = 0;
127                 break;
128             }
129         }
130         if(no_selected_child == 1){
131             if_selected = 1;
132             return;
133         }
134     }
135 }

```

그림 5 end

```

137 void SVG::enter(map<string, vector<string> > csvdata, string tag_name, int bounded_csv_index, string unique_id){
138     map<string, vector<string> >::iterator csvdata_iter;
139     if((if_selected==2) && (svg_child.size() == 0)){ //만약 scope는 자식이지만 자식이 하나도 없을 경우
140         for(csvdata_iter = csvdata.begin(); csvdata_iter!=csvdata.end(); csvdata_iter++){ //csv파일 1개의 모든 data에 대해
141             SVG *temp_SVG = new SVG(tag_name);
142             temp_SVG->Unique_Id[0] = unique_id;
143             temp_SVG->Unique_Id[1] = csvdata_iter->first;
144             temp_SVG->bounded_csv_index = bounded_csv_index;
145             svg_child.push_back(temp_SVG); // SVG추가
146         }
147         return;
148     }else{
149         int index=0;
150         for(int j=0; j<svg_child.size(); j++){
151             if(svg_child[j]->get_if_selected()==1){
152                 index=1;
153                 break;
154             }else if(svg_child[j]->get_if_selected() == 2){
155                 index=2;
156                 break;
157             }
158         }
159         if(index == 1 || index == 0){
160             for(int k=0; k<svg_child.size(); k++){
161                 svg_child[k]->if_selected = 0;
162             }
163             int if_detected = 0;
164             for(csvdata_iter = csvdata.begin(); csvdata_iter!=csvdata.end(); csvdata_iter++){
165                 if_detected = 0;
166                 for(int k=0; k<svg_child.size(); k++){
167                     if(svg_child[k]->Unique_Id[0] == unique_id){
168                         if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
169                             if(svg_child[k]->tag_name == tag_name){ //만약 이미 있는 자료일 경우 if_detected=1
170                                 if_detected = 1;
171                                 break;
172                             }
173                         }
174                     }
175                 }
176             }
177         }
178     }
179 }

```



```

174         if(if_detected == 1){ //만약 있는 자료일 경우 스킵
175             continue;
176         }else{ //없는 자료일 경우 추가
177             SVG *temp_SVG = new SVG(tag_name_);
178             temp_SVG->Unique_Id[0] = unique_id_;
179             temp_SVG->Unique_Id[1] = csvdata_iter->first;
180             temp_SVG->bounded_csv_index = bounded_csv_index_;
181             svg_child.push_back(temp_SVG);
182         }
183     }
184     return ;
185 }else if(index == 2){
186     for(int j=0; j<svg_child.size(); j++){
187         if(svg_child[j]->get_if_selected() == 2){
188             svg_child[j]->enter(csvdata_, tag_name_, bounded_csv_index_, unique_id_);
189         }
190     }
191 }
192 }
193 }
194

```

그림 7 enter

```

197 void SVG::update(map<string, vector<string> > csvdata_, string tag_name_, int bounded_csv_index_, string unique_id_){
198     map<string, vector<string> >::iterator csvdata_iter;
199     if((if_selected==2) && (svg_child.size() == 0)){ //만약 스포프는 자식인데 자식이 없으면 update를 가이므로 스킵
200         return;
201     }else{
202         int index=0;
203         for(int j=0; j<svg_child.size(); j++){
204             if(svg_child[j]->get_if_selected()==1){
205                 index=1;
206             }else if(svg_child[j]->get_if_selected() == 2){
207                 index=2;
208                 break;
209             }
210         }
211
212         if(index == 1 || index == 0){
213             for(int k=0; k<svg_child.size(); k++){
214                 svg_child[k]->if_selected = 0;
215             }
216             int if_detected = 0;
217             for(csvdata_iter = csvdata_.begin(); csvdata_iter!=csvdata_.end(); csvdata_iter++){
218                 if_detected = 0;
219                 for(int k=0; k<svg_child.size(); k++){
220                     if(svg_child[k]->Unique_Id[0] == unique_id_){
221                         if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
222                             if(svg_child[k]->tag_name == tag_name_){ //만약 있는 자료일 경우 |
223                                 svg_child[k]->bounded_csv_index = bounded_csv_index_; //bounded_csv_index를 update할 csv index를 update
224                                 svg_child[k]->if_selected = 1; //update된 객체를 선택
225                                 break;
226                             }
227                         }
228                     }
229                 }
230             }
231             return ;
232
233         }else if(index == 2){
234             for(int j=0; j<svg_child.size(); j++){
235                 if(svg_child[j]->get_if_selected()==2){
236                     svg_child[j]->update(csvdata_, tag_name_, bounded_csv_index_, unique_id_);
237                 }
238             }
239         }
240     }
241 }
242

```

그림 9 update

```

242 void SVG::exit(map<string, vector<string>> csvdata, string tag_name_, string unique_id_){ //update와 유사
243     map<string, vector<string>>::iterator csvdata_iter;
244     if((if_selected==2) && (svg_child.size() == 0)){
245         return;
246     }else{
247         int index=0;
248         for(int j=0; j<svg_child.size(); j++){
249             if(svg_child[j]->get_if_selected()==1){
250                 index=1;
251                 break;
252             }else if(svg_child[j]->get_if_selected() == 2){
253                 index=2;
254                 break;
255             }
256         }
257         if(index == 1 || index == 0){
258             for(csvdata_iter = csvdata.begin(); csvdata_iter!=csvdata.end(); csvdata_iter++){
259                 int if_detected = 0;
260                 for(int k=0; k<svg_child.size(); k++){
261                     if(svg_child[k]->Unique_Id[0] == unique_id_){
262                         if(svg_child[k]->Unique_Id[1] == csvdata_iter->first){
263                             if(svg_child[k]->tag_name == tag_name_){ //이미 있는 자료일 경우 |
264                                 svg_child[k]->if_selected = 0; //if_selected = 0
265                                 break;
266                             }
267                         }
268                     }
269                 }
270             }
271             return ;
272         }else if(index == 2){
273             for(int j=0; j<svg_child.size(); j++){
274                 if(svg_child[j]->get_if_selected()==2){
275                     svg_child[j]->exit(csvdata, tag_name_, unique_id_);
276                 }
277             }
278         }
279     }
280 }

```

그림 10 exit

```

276 void SVG::cattr(string attr_name_, string attr_value_){
277     if((if_selected == 2) && (svg_child.size() == 0)){
278         return;
279     }
280     int index=0;
281     for(int j=0; j<svg_child.size(); j++){
282         if(svg_child[j]->get_if_selected() == 1){
283             index=1;
284             break;
285         }else if(svg_child[j]->get_if_selected() == 2){
286             index=2;
287             break;
288         }
289     }
290     if(index == 1){
291         for(int j=0; j<svg_child.size(); j++){
292             if(svg_child[j]->get_if_selected()==1){
293                 svg_child[j]->attribute_iter = svg_child[j]->attribute.find(attr_name_); //attr_name이라는 attribute를 이미 가지고 있는지 체크
294                 if(svg_child[j]->attribute_iter != svg_child[j]->attribute.end()){ //있다면
295                     svg_child[j]->attribute_iter->second = attr_value_; //값 수정
296                 }else{ //없다면
297                     svg_child[j]->attribute.insert(pair<string, string>(attr_name_, attr_value_)); //속성 추가
298                 }
299             }
300         }
301         return ;
302     }else if(index == 2){
303         for(int j=0; j<svg_child.size(); j++){
304             if(svg_child[j]->get_if_selected()==2){
305                 svg_child[j]->cattr(attr_name_, attr_value_);
306                 break;
307             }
308         }
309     }else if(index == 0) return;
310 }

```

그림 11 cattr

```

312 void SVG::tattr(string x_multiplier_, string y_multiplier_){
313     if((if_selected == 2) && (svg_child.size() == 0)){
314         return;
315     }
316     int index=0;
317     for(int j=0; j<svg_child.size(); j++){
318         if(svg_child[j]->get_if_selected() == 1){
319             index=1;
320         }else if(svg_child[j]->get_if_selected() == 2){
321             index=2;
322             break;
323         }
324     }
325     if(index == 1){
326         for(int j=0; j<svg_child.size(); j++){
327             if(svg_child[j]->get_if_selected()==1){
328                 string temp_x_multiplier = x_multiplier_;
329                 string temp_y_multiplier = y_multiplier_;
330
331                 float float_x_multiplier_ = atof(x_multiplier_.c_str()) * j; //x_multiplier_ float로 바꾸고 index를 곱함
332                 float float_y_multiplier_ = atof(y_multiplier_.c_str()) * j; //y_multiplier_ float로 바꾸고 index를 곱함
333                 stringstream x_stream, y_stream;
334                 x_stream << float_x_multiplier_;
335                 temp_x_multiplier = x_stream.str();
336                 y_stream << float_y_multiplier_;
337                 temp_y_multiplier = y_stream.str();
338
339                 string str;
340                 str = "translate(";
341                 str += temp_x_multiplier;
342                 str += ", ";
343                 str += temp_y_multiplier;
344                 str += ")";
345
346                 svg_child[j]->attribute_iter = svg_child[j]->attribute.find("transform"); //transform이라는 attribute를 가지고 있는지 체크
347                 if(svg_child[j]->attribute_iter != svg_child[j]->attribute.end()){ //있다면
348                     svg_child[j]->attribute_iter -> second = str; //값 변경
349                 }else{ //없다면
350                     svg_child[j]->attribute.insert(pair<string, string>("transform", str)); //속성 추가
351                 }
352             }
353         }
354         return ;
355     }else if(index == 2){
356         for(int j=0; j<svg_child.size(); j++){
357             if(svg_child[j]->get_if_selected()==2){
358                 svg_child[j]->tattr(x_multiplier_, y_multiplier_);
359             }
360         }
361     }else if(index == 0) return;
362 }
363

```

그림 13 tattr

```

364 void SVG::dattr(vector<map<string, vector<string>>> csv_database_, vector<string> field_, string svg_attr_name_, string datum_field_name_){ //mul과 add가 주어지지 않았을 경우
365     if((if_selected == 2) && (svg_child.size() == 0)){
366         return ;
367     }
368     int index = 0;
369     if(if_selected == 2){
370         for(int i=0; i<svg_child.size(); i++){
371             if(svg_child[i]->get_if_selected() == 1) {index = 1; break;}
372             else if(svg_child[i]->get_if_selected() == 2) {index = 2; break;}
373         }
374     }
375     if(index == 1){
376         for(int i=0; i<svg_child.size(); i++){
377             if(svg_child[i]->get_if_selected() == 1){
378                 vector<map<string, vector<string>>>::iterator database_iter;
379                 vector<string>::iterator field_iter;
380
381                 svg_child[i]->attribute_iter = svg_child[i]->attribute.find(svg_attr_name_);
382
383                 if(datum_field_name_ == field_[0]){ //만약 unique id에 대해 dattr을 한다면
384                     if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
385                         svg_child[i]->attribute[svg_attr_name_] = svg_child[i]->Unique_Id[1];
386                     }else{
387                         svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, svg_child[i]->Unique_Id[1]));
388                     }
389                 }else{ //만약 attribute에 대해 한다면
390                     if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
391                         int k=0;
392                         for(k=1; k<field_.size(); k++){
393                             if(field_[k] == datum_field_name_) break; //datum_field_name이 field 중 몇 번째 인덱스 갖고 k에 저장
394                         }
395                         svg_child[i]->attribute[svg_attr_name_] = csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]][k-1]; //그곳에 해당하는 value를 저장
396                     }else{ //만약 field에서 첫 번째라면 database에서 0번째 값을 참조해야하므로 k-1로 search
397                         int k=0;
398                         for(k=1; k<field_.size(); k++){
399                             if(field_[k] == datum_field_name_) break;
400                         }
401                         svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]][k-1]));
402                     }
403                 }
404             }
405         }
406     }else continue;

```



```

407 }else if(index == 0){ //자식의 selection이 모두 0인 경우
408     return;
409 }else if(index == 2){
410     for(int i=0; i<svg_child.size(); i++){
411         if(svg_child[i]->get_if_selected() == 2){
412             svg_child[i]->datatr(csv_database_, field_, svg_attr_name_, datum_field_name_);
413         }
414     }
415 }
416 }
417
418 void SVG::datatr(vector<map<string, vector<string>>> csv_database_, vector<string> field_, string svg_attr_name_, string datum_field_name_, float mul_, float add_){
419     if((if_selected == 2) && (svg_child.size() == 0)){
420         return;
421     }
422     int index = 0;
423     if(if_selected == 2){
424         for(int i=0; i<svg_child.size(); i++){
425             if(svg_child[i]->get_if_selected() == 1) {index = 1; break;}
426             else if(svg_child[i]->get_if_selected() == 2) {index = 2; break;}
427         }
428     }
429     if(index == 1){
430         for(int i=0; i<svg_child.size(); i++){
431             if(svg_child[i]->get_if_selected() == 1){
432                 vector<map<string, vector<string>>>::iterator database_iter;
433                 vector<string>::iterator field_iter;
434                 vector<string> temp;
435                 temp = csv_database_[svg_child[i]->bounded_csv_index-1][svg_child[i]->Unique_Id[1]];
436
437                 svg_child[i]->attribute_iter = svg_child[i]->attribute.find(svg_attr_name_);
438                 if(datum_field_name_ == field_[0]){
439                     if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
440                         svg_child[i]->attribute[svg_attr_name_] = atof(svg_child[i]->Unique_Id[1].c_str()) * mul_ + add_;
441                     }else{
442                         float float_result = atof(svg_child[i]->Unique_Id[1].c_str()) * mul_ + add_;
443                         string string_result;
444                         stringstream s;
445                         s << float_result;
446                         string_result = s.str();
447                         svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, string_result)); //자본 금액을 string으로 바꾸어서 저장
448                     }
449                 }else{
450                     if(svg_child[i]->attribute_iter != svg_child[i]->attribute.end()){
451                         int k=0;
452                         for(k=1; k<field_.size(); k++){
453                             if(field_[k] == datum_field_name_) break;
454                         }
455                         float float_result = atof(temp[k-1].c_str()) * mul_ + add_;
456                         string string_result;
457                         stringstream s;
458                         s << float_result;
459                         string_result = s.str();
460                         svg_child[i]->attribute[svg_attr_name_] = string_result;
461                     }else{
462                         int k=0;
463                         for(k=1; k<field_.size(); k++){
464                             if(field_[k] == datum_field_name_) break;
465                         }
466                         float float_result = atof(temp[k-1].c_str()) * mul_ + add_;
467                         string string_result;
468                         stringstream s;
469                         s << float_result;
470                         string_result = s.str();
471                         svg_child[i]->attribute.insert(pair<string, string>(svg_attr_name_, string_result));
472                     }
473                 }
474             }else continue;
475         }
476     }else if(index == 0){
477         return;
478     }else if(index == 2){
479         for(int i=0; i<svg_child.size(); i++){
480             if(svg_child[i]->get_if_selected() == 2){
481                 svg_child[i]->datatr(csv_database_, field_, svg_attr_name_, datum_field_name_, mul_, add_);
482             }
483         }
484     }
485 }

```

그림 16 datatr

```

499 void SVG::print(FILE *html_out_){
500     fprintf(html_out_, "<%s", tag_name.c_str());
501     for(attribute_iter = attribute.begin(); attribute_iter != attribute.end(); attribute_iter++){
502         fprintf(html_out_, " %s = \"%s\"", attribute_iter -> first.c_str(), attribute_iter -> second.c_str()); //attribute print
503     }
504     fprintf(html_out_, ">");
505     for(int i=0; i<svg_child.size(); i++){ //자식에 대해 print수행
506         svg_child[i] -> print(html_out_);
507     }
508     fprintf(html_out_, "</%s>", tag_name.c_str(), if_selected);
509 }

```

그림 17 print

3)interpreter.cpp

```

1  #include <iostream>
2  #include <string>
3  #include <vector>
4  #include <map>
5  #include <fstream>
6  #include <sstream>
7  #include <algorithm>
8
9  using namespace std;
10 #include "SVG.h"
11 #include "SVG.cpp"
12
13 int main(int argc, char** argv){
14     int csv_index = 0;
15     int data_number=0;
16     vector<map<string, vector<string> > > > Csv_Database;
17     vector<string> field;
18     string unique_id;
19
20     if(argc>1){ //파일이 1개 이상 입력 되었다면
21
22         ifstream csv1File(argv[1]); //첫 번째 파일에 대해
23         if(csv1File.is_open()){
24             string csvline;
25             stringstream ss;
26             map<string, vector<string> > > csv_data;
27             map<string, vector<string> >::iterator csv_data_iter;
28
29             getline(csv1File,csvline); //첫 번째 줄 읽기
30             replace(csvline.begin(), csvline.end(), ',', ' ');
31             ss.str(csvline);
32             int check_for_key = 0;
33             string temp_line;
34             while( ss >> temp_line ){
35                 if(check_for_key == 0){
36                     field.push_back(temp_line); //field vector에 field 이름을 저장
37                     unique_id = temp_line; //첫 번째 field는 unique id이므로 unique_id에 저장
38                     check_for_key = 1;
39                 }else if(check_for_key == 1){
40                     field.push_back(temp_line);
41                 }
42             }
43
44             getline(csv1File,csvline);
45             while(getline(csv1File,csvline)){ //자료 읽기
46                 replace(csvline.begin(), csvline.end(), ',', ' ');
47                 stringstream iss(csvline);
48                 int i=0;
49                 int check_for_key = 0;
50                 string storage_for_key;
51                 while( iss >> temp_line ){
52                     if(check_for_key == 0){
53                         csv_data.insert(pair<string, vector<string> > >(temp_line, vector<string>())); //unique id 부분은 map의 key로 저장
54                         storage_for_key = temp_line;
55                         check_for_key = 1;
56                     }else if(check_for_key == 1){
57                         csv_data[storage_for_key].push_back(temp_line); //나머지는 map의 value로 저장
58                     }
59                 }
60             }
61             Csv_Database.push_back(csv_data); //csv파일 1개를 Csv_Database에 push_back
62         }

```



```

64 for(int k=2; k<argc; k++){ //나머지 파일에 대해 자료 저장
65     ifstream csv2File(argv[k]);
66     if(csv2File.is_open()){
67         string csvline;
68         map<string, vector<string> > csv_data;
69         map<string, vector<string> >::iterator csv_data_iter;
70         getline(csv2File, csvline);
71         getline(csv2File, csvline);
72         while(getline(csv2File, csvline)){
73             replace(csvline.begin(), csvline.end(), ',', ' ');
74             stringstream iss(csvline);
75             int i=0;
76             int check_for_key = 0;
77             string storage_for_key;
78             string temp_line;
79             while( iss >> temp_line ){
80                 if(check_for_key == 0){
81                     csv_data.insert(pair<string, vector<string> >(temp_line, vector<string>()));
82                     storage_for_key = temp_line;
83                     check_for_key = 1;
84                 }else if(check_for_key == 1){
85                     csv_data[storage_for_key].push_back(temp_line);
86                 }
87             }
88         }
89         Csv_Database.push_back(csv_data);
90     }
91 }
92 }
93 ////////////////////////////////////////////////////

93 ////////////////////////////////////////////////////
94 string command;
95 string storage_for_tag;
96 SVG *root_svg = new SVG("html"); //root_svg ≡ html svg 생성
97 while(true){
98     cin >> command;
99     if(command == "append"){
100         string tag_name;
101         cin >> tag_name;
102         storage_for_tag = tag_name;
103         root_svg->append(tag_name);
104     }
105     else if(command == "select"){
106         string tag_name;
107         cin >> tag_name;
108         storage_for_tag = tag_name;
109         root_svg->select(tag_name);
110     }
111     else if(command == "selectAll"){
112         string tag_name;
113         cin >> tag_name;
114         storage_for_tag = tag_name;
115         root_svg->select(tag_name);
116     }
117     else if(command == "remove"){
118         root_svg->remove();
119     }
120     else if(command == "end"){
121         if(root_svg->get_if_selected()==1) break;
122         else root_svg->end();
123     }
124     else if(command == "enter"){
125         int enter_index;
126         cin >> enter_index;
127         root_svg->enter(Csv_Database[enter_index-1], storage_for_tag, enter_index, unique_id);
128     }
}

```

```

129 else if(command == "update"){
130     int enter_index;
131     cin >> enter_index;
132     root_svg->update(Csv_Database[enter_index-1], storage_for_tag, enter_index, unique_id);
133 }
134 else if(command == "exit"){
135     int enter_index;
136     cin >> enter_index;
137     root_svg->exit(Csv_Database[enter_index-1], storage_for_tag, unique_id);
138 }
139 else if(command == "cattr"){
140     string attr_name;
141     string attr_value;
142     cin >> attr_name >> attr_value;
143     root_svg->cattr(attr_name, attr_value);
144 }
145 else if(command == "tattr"){
146     string x_multiplier;
147     string y_multiplier;
148     cin >> x_multiplier >> y_multiplier;
149     root_svg->tattr(x_multiplier, y_multiplier);
150 }
151 else if(command == "dattr"){
152     string linear_input;
153     getline(cin, linear_input);
154     string input_storage[5];
155     string svg_attr_name, datum_field_name;
156     float mul = 0.0, add = 0.0;
157     stringstream piece;
158     string piece_str;
159     int count = 0;
160     piece.str(linear_input);
161     while( piece >> piece_str){
162         input_storage[count++] = piece_str;
163     }
164     if(count == 2){ //mul과 add가 없을 때
165         svg_attr_name = input_storage[0];
166         datum_field_name = input_storage[1];
167         root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name);
168     }else if(count == 3){ //mul까지 있을 때
169         svg_attr_name = input_storage[0];
170         datum_field_name = input_storage[1];
171         mul = atof(input_storage[2].c_str());
172         add = 0.0;
173         root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name, mul, add);
174     }else if(count == 4){ //mul과 add 모두 있을 때
175         svg_attr_name = input_storage[0];
176         datum_field_name = input_storage[1];
177         mul = atof(input_storage[2].c_str());
178         add = atof(input_storage[3].c_str());
179         root_svg -> dattr(Csv_Database, field, svg_attr_name, datum_field_name, mul, add);
180     }
181 }
182 else if(command == "print"){
183     //root_svg -> print();
184     string file_name;
185     cin >> file_name;
186     FILE *html_out;
187     html_out = fopen(file_name.c_str(), "w");
188     root_svg->print(html_out);
189     fclose(html_out);
190 }
191 }
192 return 0;
193 }

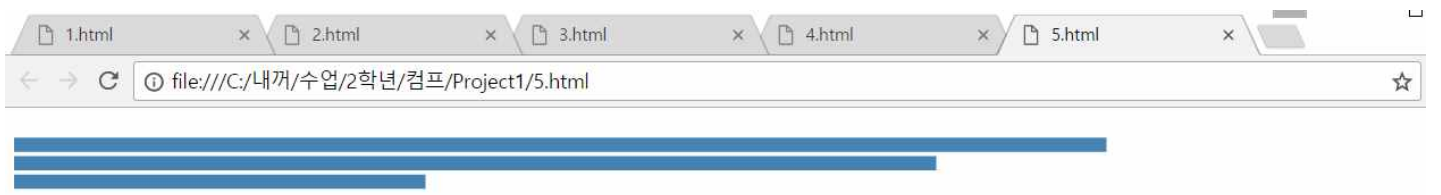
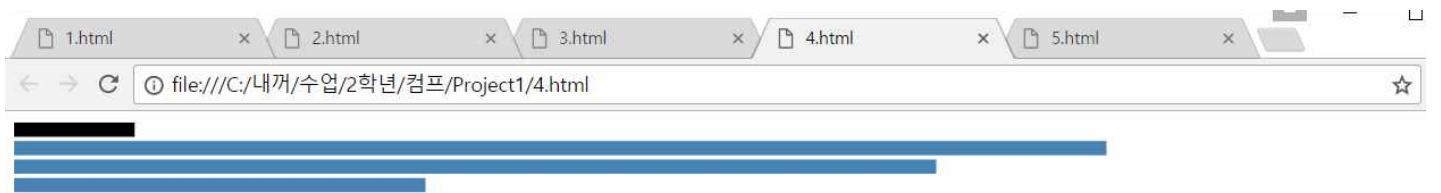
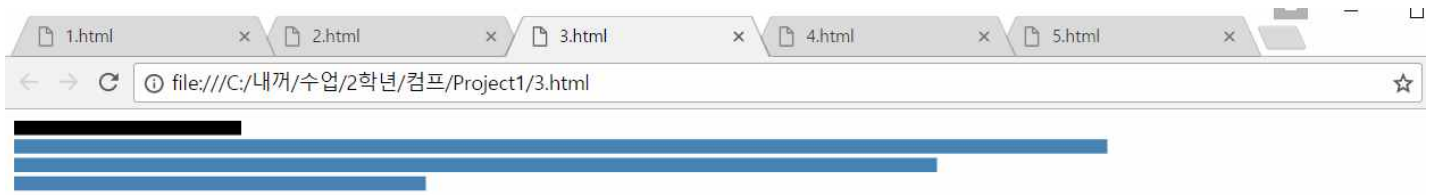
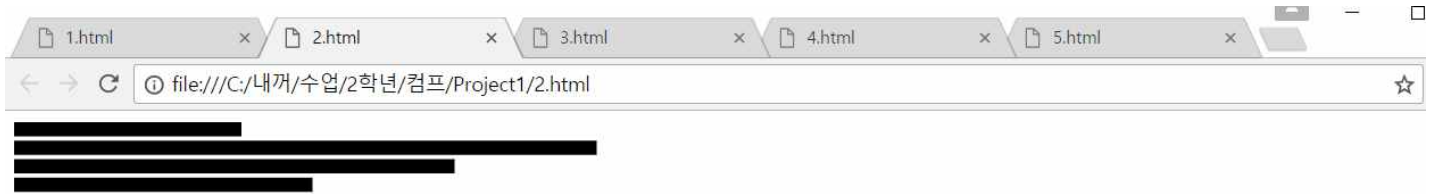
```

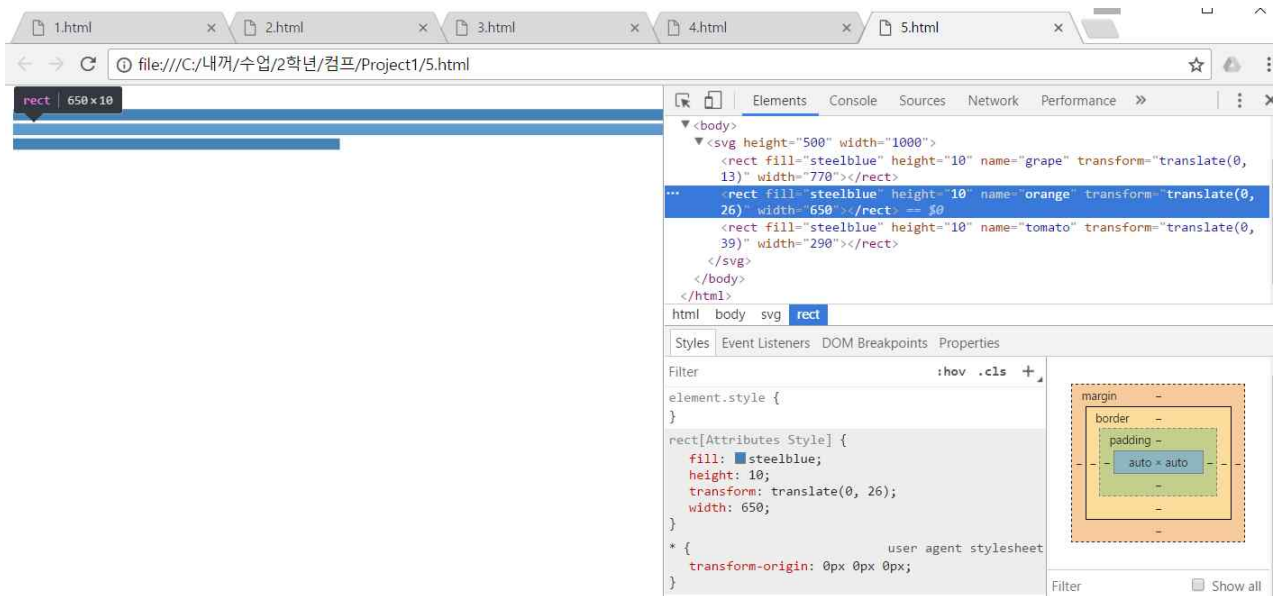
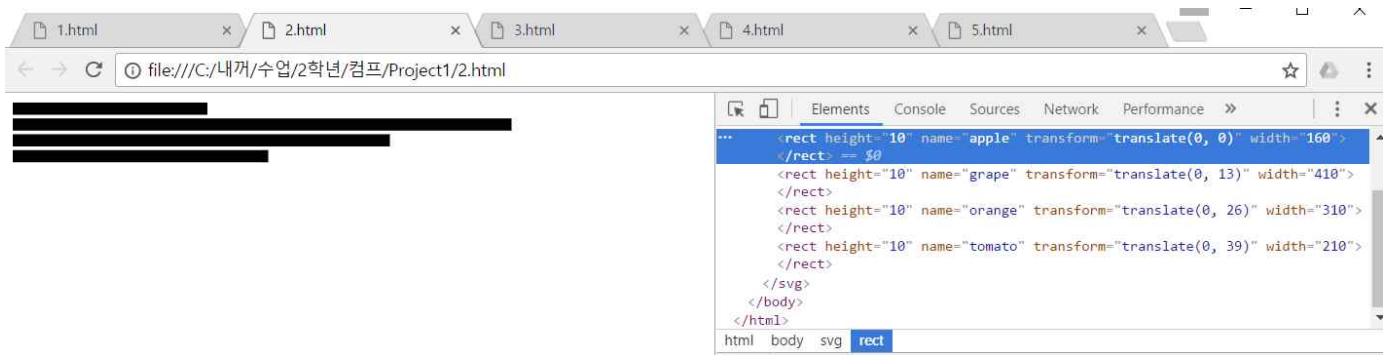
2. simulation & result

1)예제 1번

```
C:\내꺼\수업\2학년\컴프\Project1>interpreter 1.csv.txt 2.csv.txt
append svg
catr width 1000
catr height 500
print 1.html
selectAll rect
enter 1
datr name name
datr width value 0.1 10
catr height 10
tatr 0 13
print 2.html
end
selectAll rect
update 2
datr width value 0.2 10
catr fill steelblue
print 3.html
end
selectAll rect
exit 2
datr width value 0.05 10
print 4.html
remove
end
print 5.html
end
```







2)예제 2번

```
C:\내꺼\수업\2학년\컴프\Project1>interpreter 1(2).csv.txt 2(2).csv.txt
append svg
cattr width 500
cattr height 500
selectAll circle
enter 1
cattr r 3
dattr name name
dattr cx length 50
dattr cy width 50
print 1.html
end
selectAll circle
update 2
cattr fill steelblue
dattr name name
print 2.html
dattr cx length 50
dattr cy width 50
print 3.html
end
selectAll circle
exit 2
remove
end
print 4.html
end
```

