

```

1 // Ce fichier regroupe toutes les fonctions relatives aux formations
2 // JMP
3 // V1.00 - 05/04/23
4 //
5 // =====
6 // Fonction : rempli_spe_bcp()
7 // Objectif : remplir la liste déroulante avec les spécialités de Bac pro
8 //
9 // =====
10 function rempli_spe_bcp()
11 {
12 // Définition des listes 2PRO (Temps Plein / Apprentissage / Formation continue)
13 const spe_2pro_tp =
14 [["->Choisir une formation","2PRO?"],
15 ["Métiers de la Construction Durable du Bâtiment et des Travaux Publics","2PROMCDBTP"
16 ],
17 ["Métiers de la Transition Numérique et Energétique","2PROMTNE"],
18 ["Technicien Constructeur Bois","2PROTCB"],
19 ["Métiers de l'Agencement et de la Menuiserie et de l'Ameublement","2PROMAMA"],
20 ["Métiers de l'Etude et de la Modélisation Numérique Batiment","2PROMEMNB"]]
21 const spe_2pro_ap =
22 [["-> Pas de formation proposée","2PRO?"]]
23 const spe_2pro_fc =
24 [["-> Pas de formation proposée","2PRO?"]]
25 // Définition des liste 1PRO (Temps Plein / Apprentissage / Formation continue)
26 const spe_1pro_tp =
27 [["->Choisir une formation","1PRO?"],
28 ["Tech. Bat. : Organisation et réalisation du gros œuvre","1PROORGO"],
29 ["Ouvrages du Bâtiment : Métallerie","1PROMETA"],
30 ["Menuiserie Aluminium Verre","1PROMAV"],
31 ["Technicien Menuisier Agenceur","1PROTMA"],
32 ["Technicien Constructeur Bois","1PROTCB"],
33 ["Technicien Géomètre Topographe","1PROTGT"],
34 ["Tech. Etu. Bat. : Etudes et Economie","1PROTBEE"],
35 ["Tech. Etu. Bat. : Assistant en Architecture","1PROTBAA"],
36 ["Installateur en Chauffage, climatisation et Energies Renouvelables","1PROICCER"],
37 ["Maintenance et Efficacité énergétique","1PROMEE"],]
38 const spe_1pro_ap =
39 [["-> Pas de formation proposée","1PRO?"]]
40 const spe_1pro_fc =
41 [["-> Pas de formation proposée","1PRO?"]]
42 // Définition des listes TPRO (Temps Plein / Apprentissage / Formation continue)
43 const spe_tpro_tp =
44 [["->Choisir une formation","TPRO?"],
45 ["Tech. Bat. : Organisation et réalisation du gros œuvre","TPROORGO"],
46 ["Ouvrages du Bâtiment : Métallerie","TPROMETA"],
47 ["Menuiserie Aluminium Verre","TPROMAV"],
48 ["Technicien Menuisier Agenceur","TPROTMA"],
49 ["Technicien Constructeur Bois","TPROTCB"],
50 ["Technicien Géomètre Topographe","TPROTGT"],
51 ["Tech. Etu. Bat. : Etudes et Economie","TPROTBEE"],
52 ["Tech. Etu. Bat. : Assistant en Architecture","TPROTBAA"],
53 ["Installateur en Chauffage, climatisation et Energies Renouvelables","TPROICCER"],
54 ["Maintenance et Efficacité énergétique","TPROMEE"]]
55 const spe_tpro_ap =
56 [["->Choisir une formation","TPRO?"],
57 ["Installateur en Chauffage, climatisation et Energies Renouvelables","TPROICCER"],
58 ["Maintenance et Efficacité énergétique","TPROMEE"]]
59 const spe_tpro_fc =
60 [["-> Pas de formation proposée","TPRO?"]]
61 // Définition des variables
62 var sta = this.getField("eleve.statut"); // Status de l'élève
63 var spe = this.getField("bcp.specialites"); // Liste déroulante des spécialités bac
64 pro
65 var niv = this.getField("bcp.niveau"); // Niveau (2nd / 1PRO / TPRO)
66 spe.clearItems();
67 switch (niv.value)
68 {
69 case "2PRO":
70 {
71 switch (sta.value)
72 {
73 case "TP": spe.setItems(spe_2pro_tp); break;
74 case "AP": spe.setItems(spe_2pro_ap); break;
75 case "FC": spe.setItems(spe_2pro_fc); break;}
76 break;
77 }
78 }

```

```

72     case "1PRO":
73         switch (sta.value)
74         {
75             case "TP": spe.setItems(spe_lpro_tp); break;
76             case "AP": spe.setItems(spe_lpro_ap); break;
77             case "FC": spe.setItems(spe_lpro_fc); break;}
78         break;
79     case "TPRO":
80         switch (sta.value)
81         {
82             case "TP": spe.setItems(spe_tpro_tp); break;
83             case "AP": spe.setItems(spe_tpro_ap); break;
84             case "FC": spe.setItems(spe_tpro_fc); break;}
85         break;
86     }
87 }
88 // =====
89 // Fonction : afficher_famillebcp()
90 // Objectif : afficher les champs pour les secondes pro famille des métiers
91 // =====
92 function afficher_famillebcp(ast2pro)
93 {
94     // Constantes
95     const bcp2 = ["?", "1", "2"];
96     const bcp3 = ["?", "1", "2", "3"];
97     // Reset des champs concernés
98     resetForm(["elevevoeux"]);
99     // Déclaration des variables
100     var n = this.getField("bcp.niveau");
101     var t1 = this.getField("elevevoeux.bcp");
102     var v1 = this.getField("elevevoeux.bcp1");
103     var v2 = this.getField("elevevoeux.bcp2");
104     var v3 = this.getField("elevevoeux.bcp3");
105     var p1 = this.getField("elevevoeux.bcp1_priorite");
106     var p2 = this.getField("elevevoeux.bcp2_priorite");
107     var p3 = this.getField("elevevoeux.bcp3_priorite");
108
109     switch (ast2pro)
110     {
111         case "2PROMCDBTP" :
112             v1.value = "BCP ORGO";
113             v2.value = "BCP MAV";
114             v3.value = "BCP META";
115             t1.display = display.visible;
116             v1.display = display.visible;
117             v2.display = display.visible;
118             v3.display = display.visible;
119             p1.display = display.visible;
120             p1.setItems(bcp3);
121             p2.display = display.visible;
122             p2.setItems(bcp3);
123             p3.display = display.visible;
124             p3.setItems(bcp3);
125             break;
126         case "2PROMTNE" :
127             v1.value = "BCP ICCER";
128             v2.value = "BCP MEE";
129             t1.display = display.visible;
130             v1.display = display.visible;
131             v2.display = display.visible;
132             v3.display = display.hidden;
133             p1.display = display.visible;
134             p1.setItems(bcp2);
135             p2.display = display.visible;
136             p2.setItems(bcp2);
137             p3.display = display.hidden;
138             break;
139         case "2PROMEMNB" :
140             v1.value = "BCP GT";
141             v2.value = "BCP TBEE";
142             v3.value = "BCP TBAA";
143             t1.display = display.visible;
144             v1.display = display.visible;
145             v2.display = display.visible;

```

```

145     v3.display = display.visible;
146     p1.display = display.visible;
147     p1.setItems(bcp3);
148     p2.display = display.visible;
149     p2.setItems(bcp3);
150     p3.display = display.visible;
151     p3.setItems(bcp3);
152     break;
153 default :
154     t1.display = display.hidden;
155     v1.display = display.hidden;
156     v2.display = display.hidden;
157     v3.display = display.hidden;
158     p1.display = display.hidden;
159     p2.display = display.hidden;
160     p3.display = display.hidden;
161     break;
162 }
163 }
164 // =====
165 // Fonction : remplit_spe_cap()
166 // Objectif : remplir la liste déroulante avec les spécialités de CAP
167 //
168 // =====
169 function remplit_spe_cap()
170 {
171     const debug = false;
172     // Définition des listes CAP1 et CAP2 (Temps Plein / Apprentissage / Formation
    continue)
173     const spe_cap1_tp =
174     [["->Choisir une formation","CAP1?"],
175     ["Maçon","CAP1MA"],
176     ["Métallier","CAP1ME"],
177     ["Menuisier Fabricant","CAP1MN"],
178     ["Monteur en Installations Thermiques","CAP1MI"]]
179     const spe_cap1_ap =
180     [["->Choisir une formation","CAP1?"],
181     ["Métallier","CAP1ME"]]
182     const spe_cap1_fc =
183     [["-> Pas de formation proposée","CAP1?"]]
184     const spe_cap2_tp =
185     [["->Choisir une formation","CAP2?"],
186     ["Maçon","CAP2MA"],
187     ["Métallier","CAP2ME"],
188     ["Menuisier Fabricant","CAP2MN"],
189     ["Monteur en Installations Thermiques","CAP2MI"]]
190     const spe_cap2_ap =
191     [["->Choisir une formation","CAP2?"],
192     ["Métallier","CAP2ME"],
193     ["Monteur en Installations Thermiques","CAP2MI"]]
194     const spe_cap2_fc =
195     [["-> Pas de formation proposée","CAP2?"]]
196     // Définition des variables
197     var sta = this.getField("eleve.statut"); // Status de l'élève
198     var spe = this.getField("cap.specialites"); // Liste déroulante des spécialités bac
    pro
199     var niv = this.getField("cap.niveau"); // Niveau (2nd / 1PRO / TPRO)
200     spe.clearItems();
201     if (debug) {console.println(n.value);}
202     if (debug) {console.println(s.value);}
203     switch (niv.value)
204     {
205         case "CAP1" :
206             switch (sta.value)
207             {
208                 case "TP" : spe.setItems(spe_cap1_tp); break;
209                 case "AP" : spe.setItems(spe_cap1_ap); break;
210                 case "FC" : spe.setItems(spe_cap1_fc); break;}
211             break;
212         case "CAP2" :
213             switch (sta.value)
214             {
215                 case "TP" : spe.setItems(spe_cap2_tp); break;
216                 case "AP" : spe.setItems(spe_cap2_ap); break;
217                 case "FC" : spe.setItems(spe_cap2_fc); break;}

```

```

216         break;
217     break;
218     default :
219         spe.setItems(["Pb remplissage liste","PB"]);
220     }
221 }
222 // =====
223 // Fonction : remplir_spe_2nd()
224 // Objectif : remplir la liste déroulante avec les enseignements facultatifs en 2nd
225 //
226 // =====
227 function remplir_spe_2nd()
228 {const liste_spe_2nd =
229 [
230 ["-> Choisir enseignement facultatif","OPT?"],
231 ["Création et Innovation Technologique - [CIT]","CIT"],
232 ["Sciences de l'Ingénieur - [SI]","SI"]
233 ]
234 this.getField("2nd.ens_facultatif").setItems(liste_spe_2nd);
235 }
236 // =====
237 // Fonction : remplir_spe_bts()
238 // Objectif : remplir la liste déroulante avec les spécialités de BTS
239 //
240 // =====
241 function remplir_spe_bts()
242 {
243     const debug = false;
244     // Définition des listes BTS1 et BTS2 (Temps Plein / Apprentissage / Formation
    continue)
245     const spe_bts1_tp =
246     [
247     ["->Choisir une formation","BTS1?"],
248     ["Bâtiment - Temps plein sur les deux ans","BTS1BAT"],
249     ["Bâtiment - 1+1","BTS1BAT1"],
250     ["Enveloppe du Bâtiment : Conception Réalisation","BTS1EBCR"],
251     ["Fluide Energie Domotique : Genie Climatique et Fluidique","BTS1FEDA"],
252     ["Systèmes Constructif Bois et Habitats","BTS1SCBH"]
253     ]
254     const spe_bts1_ap =
255     [
256     ["->Choisir une formation","BTS1?"],
257     ["Maintenance des Systèmes Energétique et Fluidique","BTS1MSEF"],
258     ["Management Economique de la Construction","BTS1MEC"],
259     ["Métier du Géomètre Topographe et de la Modélisation Numérique","BTS1MGTMN"]
260     ]
261     const spe_bts1_fc =
262     [
263     ["->Choisir une formation","BTS1?"],
264     ["Bâtiment - 1+1","BTS1BAT1"],
265     ["Enveloppe du Bâtiment : Conception Réalisation","BTS1EBCR"],
266     ["Fluide Energie Domotique : Genie Climatique et Fluidique","BTS1FEDA"],
267     ["Management Economique de la Construction","BTS1MEC"]
268     ]
269     const spe_bts2_tp =
270     [
271     ["->Choisir une formation","BTS2?"],
272     ["Bâtiment - Temps plein sur les deux ans","BTS2BAT"],
273     ["Fluide Energie Domotique : Genie Climatique et Fluidique","BTS2FEDA"]
274     ]
275     const spe_bts2_ap =
276     [
277     ["->Choisir une formation","BTS2?"],
278     ["Bâtiment - 1+1","BTS2BAT1"],
279     ["Enveloppe du Bâtiment : Conception Réalisation","BTS2EBCR"],
280     ["Fluide Energie Domotique : Genie Climatique et Fluidique","BTS2FEDA"],
281     ["Maintenance des Systèmes Energétique et Fluidique","BTS2MSEF"],
282     ["Management Economique de la Construction","BTS2MEC"],
283     ["Métier du Géomètre Topographe et de la Modélisation Numérique","BTS2MGTMN"],
284     ["Systèmes Constructif Bois et Habitats","BTS2SCBH"]
285     ]
286     const spe_bts2_fc =
287     [
288     ["->Choisir une formation","BTS2?"],
289     ["Bâtiment - 1+1","BTS2BAT1"],
290     ["Enveloppe du Bâtiment : Conception Réalisation","BTS2EBCR"],
291     ["Fluide Energie Domotique : Genie Climatique","BTS2FEDA"],
292     ["Management Economique de la Construction","BTS2MEC"]
293     ]
294     // Définition des variables
295     var spe = this.getField("bts.specialites");
296     var niv = this.getField("bts.niveau");
297     if (debug) {console.println("->remplir_spe_bts");}
298     spe.clearItems();
299     switch (niv.value)

```

```

288 {
289     case "BTS1":
290         switch (get_statut())
291         {
292             case "TP": spe.setItems(spe_bts1_tp); break;
293             case "AP": spe.setItems(spe_bts1_ap); break;
294             case "FC": spe.setItems(spe_bts1_fc); break;}
295         break;
296     case "BTS2":
297         switch (get_statut())
298         {
299             case "TP": spe.setItems(spe_bts2_tp); break;
300             case "AP": spe.setItems(spe_bts2_ap); break;
301             case "FC": spe.setItems(spe_bts2_fc); break;}
302         break;
303     default:
304         spe.setItems(["Probleme remplissage liste","BTSPB"]);
305         break;
306 }
307 // =====
308 // Fonction : remplir_ensopt()
309 // Objectif : remplir la liste déroulante les enseignements optionnels
310 // =====
311 function remplir_ensopt(selo)
312 {const liste_ensopt_1 =
313 [
314 ["Aucun","-"],
315 ["Section Européenne Anglais en Histoire Géographie"],"EURO"]
316 ]
317 const liste_ensopt_2 =
318 [
319 ["Aucun","-"],
320 ["Section Européenne Anglais en Histoire Géographie"],"EURO"],
321 ["Section Orientale Chinois en Histoire Géographie"],"SELO"]
322 ]
323 if (selo)
324 {this.getField("ensopt.specialites").setItems(liste_ensopt_2);}
325 else
326 {this.getField("ensopt.specialites").setItems(liste_ensopt_1);}
327 resetForm("ensopt.specialites");
328 }
329 // =====
330 // Fonction : get_statut()
331 // Objectif : récupérer la valeur exportée du statut de l'élève
332 // =====
333 function get_statut()
334 {
335     return this.getField("eleve.statut").value;
336 }
337 // =====
338 // Fonction : get_voie()
339 // Objectif : récupérer la valeur exportée de la voie de formation
340 // =====
341 function get_voie()
342 {
343     var voisui = this.getField("voie_formation");
344     if (voisui.value != "Off")
345     { return voisui.value;
346     } else
347     { return "?";}
348 }
349 // =====
350 // Fonction : get_formation()
351 // Objectif : récupérer la valeur exportée de la formation
352 // =====
353 function get_formation()
354 {
355     const debug = false;
356     if (debug) {console.println("-> Get Formation");}
357     var niv;

```

```

361 var spe;
362 // Récupération de la voie suivie
363 var voisui = this.getField("voie_formationen");
364 switch (voisui.value)
365 {
366     // Voie professionnelle
367     case "LP" :
368         // La formation suivie
369         var forsui = this.getField("formations_pmf_lp");
370         if (forsui.value != "Off") { return forsui.value; } else { return "?"; }
371         break;
372     // Voie générale et technologique
373     case "LGT":
374         // La formation suivie
375         var forsui = this.getField("formations_pmf_lgt");
376         if (forsui.value != "Off") { return forsui.value; } else { return "?"; }
377     // Voie BTS
378     case "BTS":
379         var forsui = this.getField("formations_pmf_bts");
380         if (forsui.value != "Off") { return forsui.value; } else { return "?"; }
381         break;
382     default : return "?";
383 }
384 // =====
385 // Fonction : get_niveau()
386 // Objectif : Renvoyer la valeur exportée du niveau de formation
387 //
388 // =====
389 function get_niveau()
390 {
391     const debug = false;
392     if (debug) {console.println("-> get_niveau");}
393     var stniv = "NIV?";
394     switch (get_formation())
395     {
396         case "3PMET" : stniv = "3"; break;
397         case "CAP" : stniv = this.getField("cap.niveau").value; break;
398         case "BCP" : stniv = this.getField("bcp.niveau").value; break;
399         case "2ND" : stniv = "2"; break;
400         case "BACGEN" : stniv = this.getField("bacgen.niveau").value; if (stniv != "Off") {
401             stniv = stniv[0]; } break;
402         case "STI2D" : stniv = this.getField("sti2d.niveau").value; if (stniv != "Off") {
403             stniv = stniv[0]; } break;
404         case "BTS" : stniv = this.getField("bts.niveau").value; break;
405     }
406     if (debug) {console.println("<- get_niveau");}
407     return stniv;
408 }
409 // =====
410 // Fonction : get_libniveau()
411 // Objectif : Renvoyer un libellé du niveau de formation
412 //
413 // =====
414 function get_libniveau(astniv)
415 {
416     const debug = false;
417     if (debug) {console.println("-> get_libniveau");}
418     var stlibniv = "?";
419     switch (astniv)
420     {
421         case "3" : stlibniv = "3ème"; break;
422         case "CAP1" : stlibniv = "1ère année CAP"; break;
423         case "CAP2" : stlibniv = "2ème année CAP"; break;
424         case "2PRO" : stlibniv = "Seconde Bac Pro"; break;
425         case "1PRO" : stlibniv = "Première Bac Pro"; break;
426         case "TPRO" : stlibniv = "Terminale Bac Pro"; break;
427         case "2" : stlibniv = "Seconde"; break;
428         case "1" : stlibniv = "Première"; break;
429         case "T" : stlibniv = "Terminale"; break;
430         case "BTS1" : stlibniv = "1ère année BTS"; break;
431         case "BTS2" : stlibniv = "2ème année BTS"; break;
432         default : stlibniv = "NIV?"; break;
433     }
434     if (debug) {console.println("<- get_libniveau");}
435     return stlibniv;

```

```

432 }
433 // =====
434 // Fonction : get_speformation()
435 // Objectif : renvoyer la valeur exporté de la spécialité
436 //
437 // =====
438 function get_speformation()
439 {
440     const debug = false;
441     if (debug) {console.println("-> Get spe formation");}
442     var spe = "SPE?";
443     switch (get_formation())
444     {
445         case "3PMET" : spe = "PMET"; break;
446         case "CAP": if (get_niveau() != "Off") {spe = this.getField("cap.specialites").value; } else { spe="NIV?"; } break;
447         case "BCP": if (get_niveau() != "Off") {spe = this.getField("bcp.specialites").value; } else { spe="NIV?"; } break;
448         case "2ND": spe = "GT"; break;
449         case "BACGEN": if (get_niveau() != "Off") {spe = "GEN"; } else { spe="NIV?"; } break;
450         case "STI2D": if (get_niveau() != "Off") {spe = "STI2D"; } else { spe="NIV?"; } break;
451         case "BTS": if (get_niveau() != "Off") {spe = this.getField("bts.specialites").value;} else { spe="NIV?"; } break;
452         default : spe = "FOR?"; break;
453     }
454     return spe;
455 }
456 // =====
457 // Fonction : get_libformation()
458 // Objectif : renvoyer le libellé de la spécialité de la formation
459 //
460 // =====
461 function get_libformation()
462 {
463     const debug = false;
464     const lib_spe =
465     [
466         ["PMET", "MA", "MI", "ME", "MN", "MCDBTP", "MTNE", "TCB", "MAMA", "MEMNB", "ORGO", "META", "MAV",
467         "TMA", "TCB", "TGT", "TBEE", "TBAA", "ICCR", "MEE", "GT", "GEN", "STI2D", "BAT", "BAT1", "EBCR",
468         "FEDA", "SCBH", "MSEF", "MGTMN", "MEC", "?"],
469         ["3ème Prépa Métiers",
470         "Maçon",
471         "Monteur en Installations Thermiques",
472         "Métallier",
473         "Menuisier Fabricant",
474         "Métiers de la Construction Durable, du Bâtiment et des Travaux Publics",
475         "Métiers de la Transition Numérique et Energétique",
476         "Technicien Constructeur Bois",
477         "Métiers de l'Agencement et de la Menuiserie et de l'Ameublement",
478         "Métiers de l'Etude et de la Modélisation Numérique Batiment",
479         "Tech. Bat. : Organisation et réalisation du gros œuvre",
480         "Ouvrages du Bâtiment : Métallerie",
481         "Menuiserie Aluminium Verre",
482         "Technicien Menuisier Agenceur",
483         "Technicien Constructeur Bois",
484         "Technicien Géomètre Topographe",
485         "Tech. Etu. Bat. : Etudes et Economie",
486         "Tech. Etu. Bat. : Assistant en Architecture",
487         "Installateur en Chauffage, climatisation et Energies Renouvelables",
488         "Maintenance et Efficacité énergétique",
489         "Générale et Technologique",
490         "Générale",
491         "STI2D",
492         "Bâtiment - Temps sur les deux ans",
493         "Bâtiment - 1+1",
494         "Enveloppe du Bâtiment : Conception Réalisation",
495         "Fluide Energie Domotique : Genie Climatique",
496         "Systèmes Constructif Bois et Habitats",
497         "Maintenance des Systèmes Energétique et Fluidique",
498         "Métier du Géomètre Topographe et de la Modélisation Numérique",
499         "Management Economique de la Construction",
500         "Formation non définie"]],
501     if (debug) {console.println("-> Get Formation");}

```

```

498 // La formation suivie
499 var stforsui = get_formation();
500 var stniv = get_niveau();
501 var stspe = get_speformation();
502 var stlib = "";
503 if (stforsui != "?")
504 {   if (stniv != "Off")
505     {   stspe = stspe.replace(stniv, '');
506         if (lib_spe[0].indexOf(stspe) != -1)
507             {lib = get_libniveau(stniv) + " " + lib_spe[1][lib_spe[0].indexOf(stspe)];}
508         else
509             { lib = stforsui + " " + stspe; }
510     }
511     else
512     {
513         lib = stforsui + " " + "niveau de formation non défini";}
514 } else
515 { lib = "Pas de parcours de formation choisi"; }
516 if (lib === undefined) {console.println("spe : " + spe);}
517 return lib;
518 }
519 // =====
520 // Fonction : valider_spe_lg()
521 // Objectif : valider le nbr de choix des spécialités de la voie générale
522 //
523 // =====
524 function valider_spe_lg()
525 {
526     var nb_spe = 0;
527     var spe = this.getField("spegen");
528     var liste = spe.getArray();
529     for (j = 0; j < liste.length; j++) nb_spe += liste[j].isBoxChecked(0);
530     var niv = (this.getField("bacgen.niveau").value);
531     switch (niv)
532     {   case "1GEN" :
533         if (nb_spe == 3)
534             {   desactiver_spe_nock();
535                 app.alert( "Vous avez choisi les trois spécialités de première. Pour
536                     modifier votre choix, décocher les spécialités choisies");
537             } else
538             {   afficher_specialites(); }
539             break;
540         case "TGEN" :
541             if (nb_spe == 2)
542                 {   desactiver_spe_nock();
543                     app.alert( "Vous avez choisi les deux spécialités de Terminale. Pour
544                         modifier votre choix, décocher les spécialités choisies");
545                 } else
546                 {   afficher_specialites(); }
547                 break;
548         }
549     }
550 // =====
551 // Fonction : desactiver_spe_nock()
552 // Objectif : desactiver les cases à cocher spécialités de la voie générale
553 //
554 // =====
555 function desactiver_spe_nock()
556 {
557     this.getField("spegen.hggsp").readonly = (this.getField("spegen.hggsp").isBoxChecked(0)
558     )==false);
559     this.getField("spegen.llce").readonly = (this.getField("spegen.llce").isBoxChecked(0)
560     )==false);
561     this.getField("spegen.pc").readonly = (this.getField("spegen.pc").isBoxChecked(0)==
562     false);
563     this.getField("spegen.hlp").readonly = (this.getField("spegen.hlp").isBoxChecked(0)==
564     false);
565     this.getField("spegen.math").readonly = (this.getField("spegen.math").isBoxChecked(0)
566     )==false);
567     this.getField("spegen.svt").readonly = (this.getField("spegen.svt").isBoxChecked(0)==
568     false);
569     this.getField("spegen.si").readonly = (this.getField("spegen.si").isBoxChecked(0)==
570     false);

```

```

562 this.getField("spegen.nsi").readonly = (this.getField("spegen.nsi").isBoxChecked(0)==
false);
563 this.getField("spegen.ses").readonly = (this.getField("spegen.ses").isBoxChecked(0)==
false);
564 }
565 // =====
566 // Fonction : activer_ensopt()
567 // Objectif : activer le choix des enseignements optionnels
568 //
569 // =====
570 function activer_ensopt()
571 {
572 var voisui = this.getField("voie_formationen");
573 var opt = this.getField("ensopt.specialites");
574 resetForm(["ensopt.specialites"]);
575 opt.readonly = true;
576 if(voisui.value=="LGT") {opt.readonly = false;}
577 }
578 // =====
579 // Fonction : remplir_sports()
580 // Objectif : remplissage de la combo box sport.section
581 //
582 // =====
583 function remplir_sports()
584 {
585 const liste_sports = [["Rugby","RUGBY"],["Tennis","TENNIS"],["Kayak","KAYAK"]];
586 this.getField("sport.section").setItems(liste_sports);
587 }
588 // =====
589 // Fonction : activer_dispositif()
590 // Objectif : activer le choix pour le dispositif sports
591 //
592 // =====
593 function activer_dispositif()
594 {
595 this.getField("sport.section").display = display.hidden;
596 if (this.getField("sport.inscription").isBoxChecked(0)) {this.getField("sport.section")
}.display = display.visible;}
597 }
598 // =====
599 // Fonction : activer_voie
600 // Objectif : activer (readonly=true) les formations accessibles pour la voie choisie
601 //
602 // =====
603 function activer_voie()
604 {const debug = false;
605 if (debug) {console.println("->activer_voie");}
606 // Reset des formations
607 this.resetForm(["formations_pmf_lp","formations_pmf_lgt","formations_pmf_bts"]);
608 // Reset des champs des formations
609 this.resetForm(["cap","bcp","bacgen","sti2d","bts"]);
610 // Récupération de la voie demandée
611 var voisui = get_voie();
612 // Activation des formations de la voie demandée
613 this.getField("formations_pmf_lp").readonly = (voisui != "LP");
614 this.getField("formations_pmf_lgt").readonly = (voisui != "LGT");
615 this.getField("formations_pmf_bts").readonly = (voisui != "BTS");
616 if (voisui=="LGT"){this.getField("eleve.statut").value = "TP"; choix_statut();}
617 this.getField("eleve.statut").readonly = (voisui == "LGT");
618 // Activer les formations
619 activer_formation();
620 if (debug) {console.println("<-activer_voie");}
621 }
622 // =====
623 // Fonction : activer_formation()
624 // Objectif : activer les champs pour le choix de la formation
625 //
626 // =====
627 function activer_formation()
628 { const debug = false;
629 if (debug) {console.println("->activer_formation");}
630 // Remettre les valeurs par défaut des champs.
631 this.resetForm(["cap","bcp","2nd","bacgen","spegen","sti2d","bts"]);

```

```

632 // Masquer les spécialités
633 masquer_specialites();
634 var stforsi = get_formation();
635 this.getField("cap").readonly= (stforsi != "CAP");
636 this.getField("bcp").readonly= (stforsi != "BCP");
637 this.getField("2nd").readonly= (stforsi != "2ND");
638 this.getField("bacgen.niveau").readonly = (stforsi != "BACGEN");
639 this.getField("sti2d").readonly= (stforsi != "STI2D");
640 if (stforsi == "2ND")
641 {this.getField("2nd.ens_facultatif").display = display.visible;
642 } else
643 {this.getField("2nd.ens_facultatif").display = display.hidden;}
644 this.getField("bts").readonly= (stforsi != "BTS");
645 // Activer les langues
646 activer_langues();
647 // Activer enseignements optionnels
648 activer_ensopt();
649 if (debug) {console.println("<-activer_formation");}
650 }
651 // =====
652 // Fonction : masquer_specialites()
653 // Objectif : rendre invisible les champs permettant de sélectionner les spécialités
654 //
655 // =====
656 function masquer_specialites()
657 { const debug = false;
658 if (debug) {console.println(">-masquer_specialites");}
659 this.getField("cap.specialites").display = display.hidden;
660 this.getField("bcp.specialites").display = display.hidden;
661 this.getField("2nd.ens_facultatif").display = display.hidden;
662 this.getField("sti2d.ens_specifique").display = display.hidden;
663 this.getField("bts.specialites").display = display.hidden;
664 this.getField("elevevoeux").display = display.hidden;
665 this.getField("sport.section").display = display.hidden;
666 // les cases de choix pour les spécailités du bac général ne sont pas masqués mais ne
sont plus sélectionnables
667 this.getField("spegen").readonly = true;
668 if (debug) {console.println("<-masquer_specialites");}
669 }
670 // =====
671 // Fonction : afficher_specialites()
672 // Objectif : afficher les specialites
673 //
674 // =====
675 function afficher_specialites()
676 {const debug = false;
677 var cap = this.getField("cap.specialites");
678 var bcp = this.getField("bcp.specialites");
679 var sec = this.getField("2nd.ens_facultatif");
680 var sti = this.getField("sti2d.ens_specifique");
681 var bts = this.getField("bts.specialites");
682 var gen = this.getField("spegen");
683 // récupération de la formation demandée
684 var stforsi = get_formation();
685 // Masquer les spécialités
686 masquer_specialites();
687 switch (stforsi)
688 {
689 case "CAP" : rempli_spe_cap(); cap.display = display.visible; break;
690 case "BCP" : rempli_spe_bcp(); bcp.display = display.visible; break;
691 case "2ND" : sec.display = display.visible; break;
692 case "BACGEN": gen.readonly = false; break;
693 case "STI2D": sti.display = display.visible; break;
694 case "BTS" : rempli_spe_bts(); bts.display= display.visible; break;
695 default : cap.display = display.hidden; bcp.display = display.hidden; sec.display
= display.hidden; gen.readonly = true; sti.display = display.hidden; bts.display =
display.hidden;
696 }
697 }
698 // =====
699 // Fonction : rempli_lvb()
700 // Objectif : rempli la liste déroulante LVB
701 //
702 // =====

```

```

702 function remplir_lvb()
703 {
704     const liste_lvb =
705     [
706         ["ALLEMAND", "ALL"],
707         ["ESPAGNOL", "ESP"],
708         ["CHINOIS", "CHI"],
709         ["LVB-CNED", "CNED"],
710         ["LVB-AUTRE", "AUTRE"]
711     ]
712     this.getField("lv.b").setItems(liste_lvb);
713 }
714 // =====
715 // Fonction : remplir_lvc()
716 // Objectif : remplir la liste déroulante LVC
717 //
718 // =====
719 function remplir_lvc(chinois)
720 {
721     const liste_lvc_2 =
722     [
723         ["Aucune", "-"],
724         ["CHINOIS", "CHI"]
725     ]
726     const liste_lvc_1 =
727     [
728         ["Aucune", "-"]
729     ]
730     if (chinois) {this.getField("lv.c").setItems(liste_lvc_1);} else {this.getField("lv.c")
731     }.setItems(liste_lvc_2);
732 }
733 // =====
734 // Fonction : afficher_infolvb()
735 // Objectif : afficher demande info complémentaire
736 //
737 // =====
738 function afficher_infolvb(astlvb)
739 {
740     var lvb_info = this.getField("lvb.info");
741     var lvb_autre = this.getField("lvb.autre");
742     resetForm(["lvb"]);
743     switch (astlvb)
744     {
745         case "CNED" :
746             lvb_info.display = display.visible;
747             lvb_autre.display = display.visible;
748             break;
749         case "AUTRE" :
750             lvb_info.display = display.visible;
751             lvb_autre.display = display.visible;
752             break;
753         default :
754             lvb_info.display = display.hidden;
755             lvb_autre.display = display.hidden;
756             remplir_ensopt(astlvb=="CHI");
757             remplir_lvc(astlvb=="CHI");
758             break;
759     }
760 }
761 // =====
762 // Fonction : activer_langues()
763 // Objectif : activer les champs pour le choix des langues
764 //
765 // =====
766 function activer_langues()
767 {
768     // Récupération de la voie suivie
769     var voisui = this.getField("voie_formation");
770     var lvb = this.getField("lv.b");
771     var lvc = this.getField("lv.c");
772     // Réinitialisation des champs lvb et lvc
773     resetForm(["lv.b", "lv.c"]);

```

```

774 // Etat par défaut des champs langues
775 lvb.display = display.hidden;
776 lvb.readonly = false;
777 lvc.display = display.hidden;
778 switch (get_formation())
779 {
780     case "3PMET" : lvb.display = display.visible; lvb.value = "ESPAGNOL"; lvb.readonly
        = true; break;
781     case "2ND" : lvb.display = display.visible; lvc.display = display.visible; break;
782     case "BACGEN" : lvb.display = display.visible; lvc.display = display.visible;
        break;
783     case "STI2D" : lvb.display = display.visible; lvc.display = display.visible; break
        ;
784 }
785 }
786 // =====
787 // Fonction : valider_voeux(achois)
788 // Objectif : valider le numéro du voeux bac pro famille des métiers
789 //
790 // =====
791 function valider_voeux(apriorite,cbchoix1,cbchoix2)
792 {
793     var nb=0;
794     var p1 = this.getField(cbchoix1).value;
795     var p2 = this.getField(cbchoix2).value;
796     console.println(cbchoix1.name);
797     if (p1==apriorite) {resetForm(this.getField(cbchoix1).name);}
798     if (p2==apriorite) {resetForm(this.getField(cbchoix2).name);}
799 }

```