

Line wrap ☐

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="mobile-web-app-capable" content="yes">
7   <link rel="manifest" href="manifest.json">
8   <link rel="shortcut icon" type="image/png" href="favicon-512.png">
9   <title>NMS Blueprints</title>
10  <style>
11    body {background: #31b2a0 url("nms3.jpg") no-repeat fixed;}
12    #container {max-width: 703px;}
13    table {border: 0; border-collapse: collapse;}
14    td {width: 50%; vertical-align: text-top;}
15    #Localstore td {width: 100%}
16    span {display: block}
17    select:focus, input:focus {outline: none;}
18    button, select, option, input, #msg {border-radius: 8px; border: 1px solid #e0e0e0; background: rgba(240, 240, 240, 0.75)}
19    .d, .u, .m, select, option, input, #anal {font-size: 24px; font-weight: 600; margin-right: 3px; height: 50px;}
20    .d, .u {width: 50px; text-align: center;}
21    .m:not(#times) {width: 40px;}
22    .b, .d, .u, input {margin: 10px 10px 0 0;}
23    input {max-width: 350px;}
24    option {font-weight: 400;}
25    .b {padding: 5px; text-align: left; width: 97%;}
26    .r {font-weight: 600;}
27    .i {padding-left: 10px;}
28    .b:active, .m:active {background: rgba(220, 220, 220, 0.75);}
29    .u {background: rgba(240, 240, 240, 0.75) url("cloud_bw_up.png") no-repeat center;}
30    .d {background: rgba(240, 240, 240, 0.75) url("cloud_bw_down.png") no-repeat center;}
31    .u:active {background: rgba(220, 220, 220, 0.75) url("cloud_bw_up.png") no-repeat center;}
32    .d:active {background: rgba(220, 220, 220, 0.75) url("cloud_bw_down.png") no-repeat center;}
33    .c, .c:active {background: rgba(255, 255, 255, 0.75);}
34    #msg {margin:10px; font-weight: 600; padding: 5px; display: inline-block;} /* so td.width != 100% */
35    #egg {font-weight: 400; margin-left: 10px; font-style: italic; display: inline-block;}
36    #clr {
37      position: fixed; bottom:0%; margin-bottom: 10px; padding: 5px; color: white;
38      border-radius: 8px; border: 1px solid #e0e0e0; background: rgba(240, 0, 0, 0.75);
39    }
40    #clr:active {background: rgba(200, 0, 0, 0.75);}
41
42    * {font-size: 16px; font-family: Arial;}
43    @media (max-device-width:480px) {
44      /* * {font-size: 12px; font-family: Arial;} */
45      .d, .u, .m, select, option, input, #anal {font-size: 18px; font-weight: 600; margin-right: 3px; height: 50px;}
46      .m:not(#times) {width: 36px;}
47    }
48    @media (max-device-width:420px) {
49      * {font-size: 12px; font-family: Arial;}
50      .d, .u, .m, select, option, input, #anal {font-size: 18px; font-weight: 600; margin-right: 3px; height: 50px;}
51      .m:not(#times) {width: 30px;}
52    }
53    @media (max-device-width:360px) {
54      * {font-size: 12px; font-family: Arial;}
55      .d, .u, .m, select, option, input, #anal {font-size: 16px; font-weight: 600; margin-right: 3px; height: 50px;}
56      .m:not(#times) {width: 24px;}
57    }
58  </style>
59 </head>
60 <body><div id="container">
61   <div>
62     <select name="combo" id="combo"></select>
63     <button type="button" class="m" id="menu" >=</button> <!-- # -->
64     <button type="button" class="m" id="plus" >+</button>
65     <button type="button" class="m" id="times">10<sup id="power">0</sup></button>
66     <button type="button" class="m" id="ones" >1</button>
67     <button type="button" class="m" id="zero" >0</button>
68     <button type="button" class="m" id="onoff">●</button>
69     <button type="button" class="m" id="anal" >@</button>
70   </div>
71   <table id="Localstore">
72     <tr><td><input type="text" id="slot0"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
73     <tr><td><input type="text" id="slot1"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
74     <tr><td><input type="text" id="slot2"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
75     <tr><td><input type="text" id="slot3"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
76     <tr><td><input type="text" id="slot4"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
77     <tr><td><input type="text" id="slot5"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
78     <tr><td><input type="text" id="slot6"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
79     <tr><td><input type="text" id="slot7"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
80     <tr><td><input type="text" id="slot8"><button class="u">&nbsp;</button><button class="d">&nbsp;</button></td></tr>
81     <tr><td><div id="msg">To delete: clear entry then click upload<div></td></tr>
82   </table>
83   <table id="Components"><tr><td class="col"></td><td class="col"></td></tr></table>
84   </div>
85   <button id="clr">Click to clear "localStorage"</button>
86   <div id="info" style="display:block;font-size:10px;width:30px;margin-top:10px;padding-left:5px;"></div>
87   <script>
88     let cp, rpt, dblcol = getQueryVariable('dblcol'),
89     db = Object.create(null),
90     price = Object.create(null),
91     polestar = Object.create(null),
92     include = Object.create(null), // used in displayCp() to exclude some categories
93     combo = document.getElementById("combo"),
94     anal = document.getElementById("anal");
95     combo.addEventListener("change", () => {
96       changeDb(combo.value)
97     });
```

```

98 anal.addEventListener("click", () => {
99     document.getElementById("anal").style.display = "none";
100     document.getElementById("clr").style.display = "none";
101     let doc = document.querySelectorAll("#Components .col"),
102     status = document.getElementById("menu").innerHTML;
103     doc[0].innerHTML = "";
104     doc[1].innerHTML = "";
105     analyse("Stasis Device", status, doc[0]);
106     analyse("Fusion Ignitor", status, doc[1]);
107 });
108 document.getElementById("clr").addEventListener("click", () => {
109     let doc = document.querySelectorAll('input[id^=slot]');
110     for (let i = 0; i < doc.length; i++) {
111         localStorage.removeItem(`slot${i}`);
112         document.getElementById(`slot${i}`).value = "";
113     }
114     document.getElementById("clr").style.display = "none";
115 });
116 document.getElementById("menu").addEventListener("click", toggle_menu);
117 document.getElementById("plus").addEventListener("click", toggle_plus);
118 document.getElementById("times").addEventListener("click", toggle_times);
119 document.getElementById("ones").addEventListener("click", toggle_ones);
120 document.getElementById("zero").addEventListener("click", toggle_zero);
121 document.getElementById("onoff").addEventListener("click", toggle_onoff);
122 addCombo("Components");
123 init();
124 addCombo("Localstore");
125 displayDb();
126 document.querySelector('#combo [value="CrashedShip"]').selected = true;
127 changeDb("CrashedShip");
128 console.log(price);
129 console.log(db);
130 document.getElementById("info").innerHTML = screen.width;
131
132 // storing an item in slot7 called "Polestar" not allowed
133 let slot7 = localStorage.getItem("slot7");
134 if(slot7){
135     let [polestar] = JSON.parse(slot7);
136     if(polestar == "Polestar"){
137         localStorage.removeItem("slot7");
138     }
139 }
140 // refresh "All Ones"
141 let slot8 = localStorage.getItem("slot8");
142 if(slot8){
143     let [polestar] = JSON.parse(slot8);
144     if(polestar == "All Ones"){
145         localStorage.removeItem("slot8");
146     }
147 }
148 /* create a default 'All Ones' in localStorage[slot8] */
149 if(localStorage.getItem("slot8") === null) {
150     let ones = JSON.parse(JSON.stringify(db));
151     for(const cat in db){
152         for(const prop in db[cat]){
153             ones[cat][prop][0] = 1;
154         }
155     }
156     localStorage.setItem('slot8', JSON.stringify(['All Ones', ones]));
157 }
158
159 /* add event listeners for Localstore buttons */
160 let btn = document.querySelectorAll("#Localstore button");
161 for (let i = 0; i < btn.length; i++) {
162     btn[i].addEventListener("click", localstore);
163 }
164
165 /* populate localstore input fields with descriptions */
166 btn = document.querySelectorAll("#Localstore .d");
167 for (let i = 0; i < btn.length; i++) {
168     let slot = localStorage.getItem(`slot${i}`), desc = '';
169     if(slot){
170         [desc] = JSON.parse(slot);
171     }
172     document.getElementById(`slot${i}`).value = desc;
173 }
174
175 /* event handler for Localstore */
176 function localstore(e){
177     console.log("localstore");
178     let text = e.target.parentElement.firstChild,
179     key = e.target.parentElement.firstChild.id,
180     op = e.target.className;
181     console.log(text, key, op);
182     if(op == "u"){
183         if(text.value.match(/^\s*$/)){
184             localStorage.removeItem(key);
185             hilite();
186         }else{
187             if(!(key == "slot8" && text.value == "All Ones")){ // can't alter slot8:"All Ones"
188                 localStorage.setItem(key, JSON.stringify([text.value, db]));
189                 hilite(key);
190             }
191         }
192     }else{
193         let data = localStorage.getItem(key);
194         if(data){
195             [desc, data] = JSON.parse(data);
196             // text.value = desc;
197             hilite(key);
198         }
199     }
200 }

```

```

197     db = data;
198     displayDb();
199 }
200 }
201 }
202
203 /* hilite recently accessed localstore slot */
204 function hilite(slot){
205     let inp = document.querySelectorAll("#Localstore input");
206     for (let i = 0; i < inp.length; i++) {
207         inp[i].style.backgroundColor = inp[i].id == slot ? "rgba(220, 220, 220, 0.9)" : "rgba(240, 240, 240, 0.75)";
208     }
209 }
210
211 /* set increment amount to 1, 10, 100, 1000 */
212 function toggle_times(){
213     let text = document.getElementById("power");
214     text.innerHTML = (parseInt(text.innerHTML) + 1) % 4;
215 }
216
217 /* toggle blueprint between summary and detail */
218 function toggle_menu(){
219     let text = document.getElementById("menu"), attr;
220     [text.innerHTML, attr] = text.innerHTML == "=" ? ["#", "none"] : ["=", "block"];
221     let elements = document.getElementsByClassName('i');
222     for (let i = 0; i < elements.length; i++) {
223         elements[i].style.display = attr;
224     }
225     flow();
226 }
227
228 /* toggle whether to add or subtract increment amount */
229 function toggle_plus(){
230     let text = document.getElementById("plus");
231     text.innerHTML = text.innerHTML == "+" ? "-" : "+";
232 }
233
234 /* set all amounts to 1 in current category */
235 function toggle_ones(){
236     let cat = document.getElementById("combo").value;
237     for(const prop in db[cat]){
238         db[cat][prop][0] = 1;
239         document.getElementById(prop).childNodes[0].innerHTML = `${prop},${db[cat][prop][0]}`;
240     }
241 }
242
243 /* set all amounts to 0 in current category */
244 function toggle_zero(){
245     let cat = document.getElementById("combo").value;
246     for(const prop in db[cat]){
247         db[cat][prop][0] = 0;
248         document.getElementById(prop).childNodes[0].innerHTML = `${prop},${db[cat][prop][0]}`;
249     }
250 }
251
252 /* toggle whether this category is included in components view */
253 function toggle_onoff(){
254     let text = document.getElementById("onoff"),
255         cat = document.getElementById("combo").value;
256     include[cat] = text.innerHTML = text.innerHTML == "o" ? "●" : "o";
257 }
258
259 /* change category to display */
260 function changeDb(cat){
261     console.log(`cat=${cat}`);
262     let doc = document.querySelectorAll("table");
263     for (let i = 0; i < doc.length; i++) {
264         doc[i].style.display = 'none';
265     }
266     document.getElementById("info").style.display = "none";
267     document.getElementById("clr").style.display = "none";
268     document.getElementById(cat).style.display = 'block';
269     doc = document.querySelectorAll(".m"); // menu buttons
270     for (let i = 0; i < doc.length; i++) {
271         doc[i].style.display = 'none';
272     }
273     if(cat == "Localstore"){
274         document.getElementById("info").style.display = "block";
275     }else if(cat == "Components"){
276         document.getElementById("menu").style.display = "inline-block";
277         displayCp();
278         flow();
279     }else if(cat != "Localstore"){
280         document.getElementById("menu").style.display = "inline-block";
281         document.getElementById("plus").style.display = "inline-block";
282         document.getElementById("times").style.display = "inline-block";
283         document.getElementById("ones").style.display = "inline-block";
284         document.getElementById("zero").style.display = "inline-block";
285         document.getElementById("onoff").style.display = "inline-block";
286         document.getElementById("onoff").innerHTML = include[cat];
287         flow();
288     }
289 }
290
291 /* evenly balance the buttons in the two .col divs */
292 function flow(){
293     let cat = combo.options[combo.selectedIndex].text,
294         status = document.getElementById("menu").innerHTML,
295         tbl = document.getElementById(cat),
296         doc = tbl.querySelectorAll(".col"),
297         arrR = JSON.parse(tbl.dataset.col),
298         arrL = arrR.splice(0, doc[0].children.length);
299     let direction = delta(arrL, arrR, status);

```

```

296 | if(dblcol == "off"){
297 |     doc[1].style.display = "none";
298 |     return;
299 | }
300 | while( direction ){
301 |     if(direction > 0){
302 |         doc[0].appendChild(doc[1].firstChild);
303 |         arrL.push(arrR.shift());
304 |     }else{
305 |         doc[1].prepend(doc[0].lastChild);
306 |         arrR.unshift(arrL.pop());
307 |     }
308 |     direction = delta(arrL,arrR,status);
309 | }
310 | }
311 |
312 | function delta(aL, aR, status){
313 |     let pad = (n) => `${n.toFixed(2)}`.padStart(6, ' '),
314 |         factor = (n, status) => 40/18 + (status == "=" ? n : 0),
315 |         sumL = aL.reduce((a, b) => a + factor(b, status), 0),
316 |         sumR = aR.reduce((a, b) => a + factor(b, status), 0),
317 |         delta1 = Math.abs(sumR - sumL),
318 |         delta0 = null,
319 |         delta2 = null;
320 |     if(aL.length > 0)
321 |         delta0 = Math.abs(sumR - sumL + 2 * (factor(aL[aL.length-1], status)));
322 |     if(aR.length > 0)
323 |         delta2 = Math.abs(sumR - sumL - 2 * (factor(aR[0], status)));
324 |     console.log(`${status}, (${aL.length})[${aL}]=${pad(sumL)}, (${aR.length})[${aR}]=${pad(sumR)}, ` +
325 |         `d0=${aL.length > 0?pad(delta0):delta0}, d1=${pad(delta1)}, d2=${aR.length > 0?pad(delta2):delta2}`);
326 |     if(sumR == 0){
327 |         if(aL.length > 1){
328 |             return -1;
329 |         }else{
330 |             return 0;
331 |         }
332 |     }
333 |     if(sumL == 0){
334 |         if(aR.length > 1){
335 |             return +1;
336 |         }else{
337 |             return 0;
338 |         }
339 |     }
340 |     if(sumL + .001 > sumR){
341 |         if(delta1 > delta0 + .001) {
342 |             return -1;
343 |         }
344 |     }
345 |     if(sumR + .001 > sumL){
346 |         if(delta1 + .001 > delta2) {
347 |             return +1;
348 |         }
349 |     }
350 |     return 0;
351 | }
352 | /* delta2([],JSON.parse(document.getElementById("CrashedShip").dataset.col),'=') */
353 | function delta2(aL, aR, status){
354 |     function fact(n){return 40/18 + (status == "=" ? n : 0)}
355 |     function summ(array){return array.reduce((a, b) => a + fact(b, status), 0)}
356 |     let delta = [];
357 |     while( true ){
358 |         delta.push(Math.abs(summ(aR) - summ(aL)));
359 |         aL.push(aR.shift());
360 |         if(!aR.length) break;
361 |     }
362 |     let minIndex = delta.reduce((before, value, index, array) => value < array[before] ? index : before, 0);
363 |     console.log(minIndex,delta);
364 | }
365 |
366 | /* populate all categories with blueprints */
367 | function displayDb(){
368 |     let status = document.getElementById("menu").innerHTML, element;
369 |     for(const cat in db){
370 |         let arrL = [], arrR = [];
371 |         element = document.getElementById(cat);
372 |         if(element) element.remove();
373 |         let tbl = document.createElement("table");
374 |         tbl.setAttribute("id", cat);
375 |         tbl.style.display = "none";
376 |         tbl.innerHTML = `<tr><td class="col"></td><td class="col"></td></tr>`;
377 |         document.getElementById("container").appendChild(tbl);
378 |         let doc = document.querySelectorAll(`#${cat} .col`);
379 |
380 |         for(const prop in db[cat]){
381 |             element = document.createElement("button");
382 |             element.setAttribute("id", prop);
383 |             element.setAttribute("class", "b");
384 |             createSpan(element, "r", "=", `${prop},${db[cat][prop][0]}`);
385 |             arrR.push(0);
386 |             db[cat][prop].forEach((e, i) => {
387 |                 if(i > 0){ // skip counter entry
388 |                     createSpan(element, "i", status, e);
389 |                     arrR[arrR.length-1] += 1;
390 |                 }
391 |             });
392 |             doc[0].appendChild(element);
393 |             //console.log(prop,element);
394 |             document.getElementById(prop).addEventListener("click", accum);

```

```

395     }
396     tbl.setAttribute("data-col", `${arrR}`);
397     console.log(cat, tbl.dataset.col);
398 }
399 }
400 /* calculate component view */
401 function displayCp(){
402     cp = Object.create(null);
403     let doc = document.querySelectorAll(`#Components .col`),
404         status = document.getElementById("menu").innerHTML,
405         cost = 0, num, element, key, value, arrL = [], arrR = [];
406
407     doc[0].innerHTML = "";
408     doc[1].innerHTML = "";
409
410     for(const cat in db){
411         for(const prop in db[cat]){
412             db[cat][prop].forEach((e, i) => {
413                 if(i == 0){ // save counter in num
414                     num = e;
415                 } else {
416                     if(num > 0 && include[cat] == "●"){
417                         [key, value] = e.split(/,/);
418                         if(key != ""){ // not a dependency
419                             if(!(key in cp)){
420                                 cp[key] = Object.create(null);
421                                 cp[key].num = 0;
422                                 cp[key].grp = [];
423                             }
424                             cp[key].num += ( !prop.match(/^(\$\\)/) && key.match(/^_/) ? 1 : num ) * value;
425                             cp[key].grp.push(`x${num} ${prop},${value}`);
426                         } } } } }
427     console.log(cp);
428
429     Object.keys(cp).sort().forEach((prop) => {
430         element = document.createElement("button");
431         element.setAttribute("class", "b c");
432         createSpan(element, "r", "=", `${prop},${cp[prop].num} @${price[prop]}`);
433         cost += cp[prop].num * price[prop];
434         arrR.push(0);
435         cp[prop].grp.forEach((text, i) =>{
436             createSpan(element, "i", status, text);
437             arrR[arrR.length-1] += 1;
438         });
439         doc[0].appendChild(element);
440     });
441
442     element = document.createElement("button");
443     element.setAttribute("class", "b c");
444     createSpan(element, "r", "=", `cost: ${cost} ${cost == 0 ?
445         '<span id="egg">Click "@" to see easter egg</span>' : ""}`);
446     doc[0].appendChild(element);
447     arrR.push(0);
448
449     let tbl = document.getElementById("Components");
450     tbl.setAttribute("data-col", `${arrR}`);
451     console.log("Components", tbl.dataset.col);
452
453     if(cost == 0){
454         document.getElementById("anal").style.display = "inline-block";
455         document.getElementById("clr").style.display = "inline-block";
456     }
457 }
458 /* analyze valuable blueprints */
459 function analyse(recipe,status,doc){
460     let ref = db.Valuables[recipe];
461     for(let i = 1; i < ref.length; i++){
462         let [data,cnt] = ref[i].split(/,/);
463         if(data in db.Valuables){
464             analyse(data,status,doc);
465         }
466     }
467     //console.log(JSON.stringify([recipe,ref]));
468     let element = document.createElement("button");
469     element.setAttribute("class", "b c");
470     createSpan(element, "r", "=", `${recipe},${recipe == "Glass" ? 5 : 1}`);
471     ref.forEach((text, i) =>{
472         if(i > 0) createSpan(element, "i", status, text);
473     });
474     doc.appendChild(element);
475 }
476
477 function createSpan(element, c, status, text){
478     rpt += c == "r" ? `\\n${text}\\n` : ` ${text}\\n`;
479     //console.log(c,status,text);
480     let span = document.createElement("span");
481     span.setAttribute("class", c);
482     span.style.display = status == "=" ? "block" : "none";
483     span.innerHTML = text;
484     element.appendChild(span);
485 }
486 /* accumulator used when a blueprint is clicked */
487 function accum(e){
488     let text = document.getElementById("plus").innerHTML,
489         power = parseInt(document.getElementById("power").innerHTML),
490         element = e.target.nodeName == "SPAN" ? e.target.parentNode : e.target,
491         cat = element.parentElement.parentElement.parentElement.id,
492         num = db[cat][element.id][0];
493     //console.log(num, power,10**power);
494     if(text == "+"){

```

```

495     db[cat][element.id][0] = Math.min(db[cat][element.id][0] + 10 ** power, 10000);
496 } else {
497     db[cat][element.id][0] = Math.max(db[cat][element.id][0] - 10 ** power, 0);
498 }
499 element.childNodes[0].innerHTML = `${element.id},${db[cat][element.id][0]}`;
500 // console.log(cat,element.id,db[cat][element.id]);
501 }
502
503 /* https://shirha.github.io/blueprints/?dblcol=off */
504 function getQueryVariable(variable){
505     let query = window.location.search.substring(1);
506     let vars = query.split("&");
507     for (let i=0;i<vars.length;i++) {
508         let pair = vars[i].split("=");
509         if(pair[0] == variable){return pair[1];}
510     }
511     return(false);
512 }
513
514 /* used by init() to populate select options */
515 function addCombo(textb) {
516     var combo = document.getElementById("combo"),
517         option = document.createElement("option");
518     option.text = textb;
519     option.value = textb;
520     combo.add(option, null);
521     include[textb] = "●";
522 }
523
524 /* initialise 'price' and 'db' with blueprints */
525 function init(){
526     let cat, key, value;
527     `Acid,188000
528     Activated Copper,75
529     Amino Chamber,12300
530     Ammonia,62
531     Antimatter,5233
532     Antimatter Housing,6500
533     Aronium,25000
534     Cactus Flesh,28
535     Cadmium,83
536     Carbon,12
537     Carbon Nanotubes,500
538     Condensed Carbon,24
539     Chlorine,205
540     Chromatic Metal,88
541     Circuit Board,916250
542     Cobalt,76
543     Copper,41
544     Counterfeit Circuits,68000
545     Creature Pellets,20
546     Cryo-Pump,1500000
547     Cryogenic Chamber,3800000
548     Crystal Sulphide,7410
549     Cyto-Phosphate,201
550     Deuterium,34
551     Di-hydrogen,34
552     Di-hydrogen Jelly,200
553     Dioxide,62
554     Dirty Bronze,25000
555     Emeril,114
556     Enriched Carbon,50000
557     Faecium,30
558     Ferrite Dust,14
559     Frost Crystal,12
560     Fungal Mould,16
561     Fusion Accelerant,1500000
562     Gamma Root,16
563     Geodesite,150000
564     Glass,200
565     Gold,353
566     Grantine,25000
567     Gravitino Ball,40023
568     Heat Capacitor,180000
569     Hermetic Seal,800
570     Herrox,25000
571     Hot Ice,320000
572     Hydraulic Wiring,3600
573     Indium,132
574     Ion Battery,200
575     Ionised Cobalt,162
576     Iridesite,150000
577     Korvax Convergence Cube,13063
578     Larval Core,69875
579     Lemmium,25000
580     Life Support Gel,200
581     Liquid Explosive,800500
582     Living Glass,566000
583     Living Pearl,4797
584     Lubricant,110000
585     Magnetic Resonator,6150
586     Magnetised Ferrite,82
587     Mango-Gold,25000
588     Metal Plating,800
589     Microprocessor,2000
590     Mordite,40
591     _Nanites,0
592     NipNip Buds,17776
593     Nitrogen,20

```

```

593 Nitrogen Salt,50000
594 Organic Catalyst,320000
595 Oxygen,34
596 Paraffinium,62
597 Phosphorus,62
598 Platinum,505
599 Poly Fibre,130000
600 Portable Reactor,4200000
601 Projectile Ammunition,1
602 Pugneum,138
603 Pure Ferrite,28
604 Pyrite,62
605 Quad Servo,20000
606 Quantum Computer,4200
607 Quantum Processor,4400000
608 Radon,20
609 Salt,101
610 _Salvaged Data,0
611 _Salvaged Frigate Mod,0
612 Semiconductor,320000
613 Silicate Powder,2
614 Silver,186
615 Sodium,41
616 Sodium Nitrate,82
617 Solar Mirror,6150
618 Solanium,70
619 Star Bulb,32
620 Storm Crystal,211150
621 Sulphurine,20
622 Superconductor,1500000
623 Thermic Condensate,50000
624 Tritium,6
625 Tritium Hypercluster,3280
626 Unstable Gel,50000
627 Unstable Plasma,5750
628 Uranium,62
629 Walker Brain,35000
630 Warp Cell,46750
631 Warp Hypercore,46750
632 Wiring Loom,25000
633 .split(/\n/).forEach((e) => {
634     const [key, value] = e.split(/,/);
635     price[key] = value;
636 });
637 `CrashedShip
638   (R)Deflector Shield
639     Chromatic Metal,50
640     Sodium Nitrate,13
641
642   (R)Efficient Thrusters
643     Tritium,100
644     Wiring Loom,2
645     Di-hydrogen Jelly,1
646
647   (R)Emergency Warp Unit
648     Warp Cell,1
649     Gold,110
650     Microprocessor,2
651
652   (R)Hyperdrive
653     Chromatic Metal,63
654     Microprocessor,3
655
656   (R)Launch Sys Recharger
657     Antimatter,3
658     Magnetic Resonator,1
659     Wiring Loom,1
660
661   (R)Launch Thrusters
662     Pure Ferrite,50
663     Di-hydrogen Jelly,1
664
665   (R)Photon Cannon
666     Pure Ferrite,50
667     Sodium Nitrate,30
668
669   (R)Pulse Engine
670     Hermetic Seal,1
671     Metal Plating,1
672
673   Burnt-Out Compressor
674     Wiring Loom,2
675     Antimatter,3
676
677   Containment Failure
678     Magnetised Ferrite,50
679     Chromatic Metal,60
680     Gold,45
681
682   Corroded Tanks
683     Activated Copper,75
684
685   Damaged Gears
686     Chromatic Metal,150
687     Wiring Loom,3
688
689   Exploded Panel
690     Pure Ferrite,125
691     Gold,75
692
693   Hull Fracture
694     Chromatic Metal,100

```

```

2679     Fungal Mould,50
2680     Ammonia,25
2681
2682     Venom Urchin
2683     _Salvaged Data,3
2684     Emeril,100
2685     Gold,50
2686
2687     Frostwort
2688     _Salvaged Data,3
2689     Frost Crystal,50
2690     Dioxide,25
2691
2692     Gravitino Host
2693     _Salvaged Data,3
2694     Chromatic Metal,25
2695     Silver,120
2696     Magnetised Ferrite,25
2697
2698     Gamma Weed
2699     _Salvaged Data,3
2700     Gamma Root,50
2701     Uranium,25
2702
2703     Gutrot Flower
2704     _Salvaged Data,3
2705     Faecium,40
2706
2707     Echinocactus
2708     _Salvaged Data,3
2709     Cactus Flesh,50
2710     Pyrite,25
2711
2712     Mordite Root
2713     _Salvaged Data,3
2714     Mordite,40
2715
2716     Star Bramble
2717     _Salvaged Data,3
2718     Star Bulb,50
2719     Paraffinium,25
2720
2721     Albumen Pearl Orb
2722     _Salvaged Data,3
2723     Indium,60
2724     Paraffinium,20
2725
2726     Solar Vine
2727     _Salvaged Data,3
2728     Solanium,50
2729     Phosphorus,25
2730
2731     NipNip
2732     _Salvaged Data,3
2733     NipNip Buds,1
2734     Faecium,20`
2735     .split(/\n/).forEach((e) => {
2736         if(e && !e.match(/^\\s*$/)){
2737             if(e.match(/^\\S/)){
2738                 cat = e;
2739                 addCombo(e);
2740             }else if(e.match(/^ {4}/)){
2741                 db[cat][key].push(e.trim());
2742             }else{
2743                 key = e.trim()
2744                 if(!(cat in db)){
2745                     db[cat] = Object.create(null);
2746                 }
2747                 db[cat][key] = [0]; // accumulator 1st item in array
2748             }
2749         }
2750     });
2751     polestar = {"Freighter":{"(R)Atmospheric Control Unit":[1,"Chromatic Metal,60","Gold,30","Wiring Loom,1"],"(R)Freighter Hyperdrive":[1,"Chromatic Metal
2752     ]
2753     /*
2754     listeners
2755     *function combo[change]
2756     *function anal[click]
2757     function analyse(recipe,status,doc){
2758
2759     *function localstore[click]
2760     *function clr[click]
2761     function localstore(e){
2762     function hilite(slot){
2763
2764
2765     function toggle_menu(){
2766     function toggle_plus(){
2767     function toggle_times(){
2768     function toggle_ones(){
2769     function toggle_zero(){
2770     function toggle_onoff(){
2771
2772
2773     function flow(){
2774     function delta(aL, aR, status){
2775
2776     function changeDb(cat){
2777     function displayDb(){
2778     function displayCp(){
2779     function createSpan(element, c, status, text){
2780     function accum(e){
2781
2782     function addCombo(textb) {

```



```
2783 function init(){
2784
2785 > ipconfig
2786 Ethernet adapter Ethernet 2:
2787
2788     Connection-specific DNS Suffix  . : hsd1.ca.comcast.net. <===== look for comcast
2789     IPv6 Address. . . . . : 2601:204:e000:4dd0::4
2790     IPv6 Address. . . . . : 2601:204:e000:4dd0:952a:5e2b:5ac2:44a9
2791     Temporary IPv6 Address. . . . . : 2601:204:e000:4dd0:d033:1714:c047:e8b6
2792     Link-local IPv6 Address . . . . . : fe80::952a:5e2b:5ac2:44a9%31
2793     IPv4 Address. . . . . : 192.168.0.10 <===== 192.168.0.10:9001/
2794     Subnet Mask . . . . . : 255.255.255.0
2795     Default Gateway . . . . . : fe80::d60a:a9ff:fea2:af2c%31
2796                                192.168.0.1
2797 */
2798 </script>
2799 </body>
2800 </html>
```