

BODYFIT

SHAPE YOUR BODY



SHAPE YOUR BODY



Purpose

CUSTOMISED PLAN JUST FOR YOU

The BodyFit applications main purpose is to provide the end user with a custom meal and fitness plan. This is provided based on the users input. It takes into consideration their goals, BMI and current exercise levels.

GYM BODYFIT GYM BODYFIT GYM BODYFIT GYM BODYFIT



B
R
O
C
E
S
E
S
S
S

User analysis

Planning

Feature outline

Testing each feature in a separate doc

Bringing it all together

Error handling

Test, test, test

Features

FEATURES THAT DELIVER AN PURPOSEFUL OUTCOME

There are several features of the terminal application.

There is the ability for the end user to select their goal and this provides a motivational quote around their goal, keeping them engaged with the application.

Another feature calculates their BMI and lets then know if this is currently a healthy weight based off their gender, height and weight.

It then further provides an overall mean and exercise plan based off all their various selections.



The CODE

KEEPING IT DRY

CODE

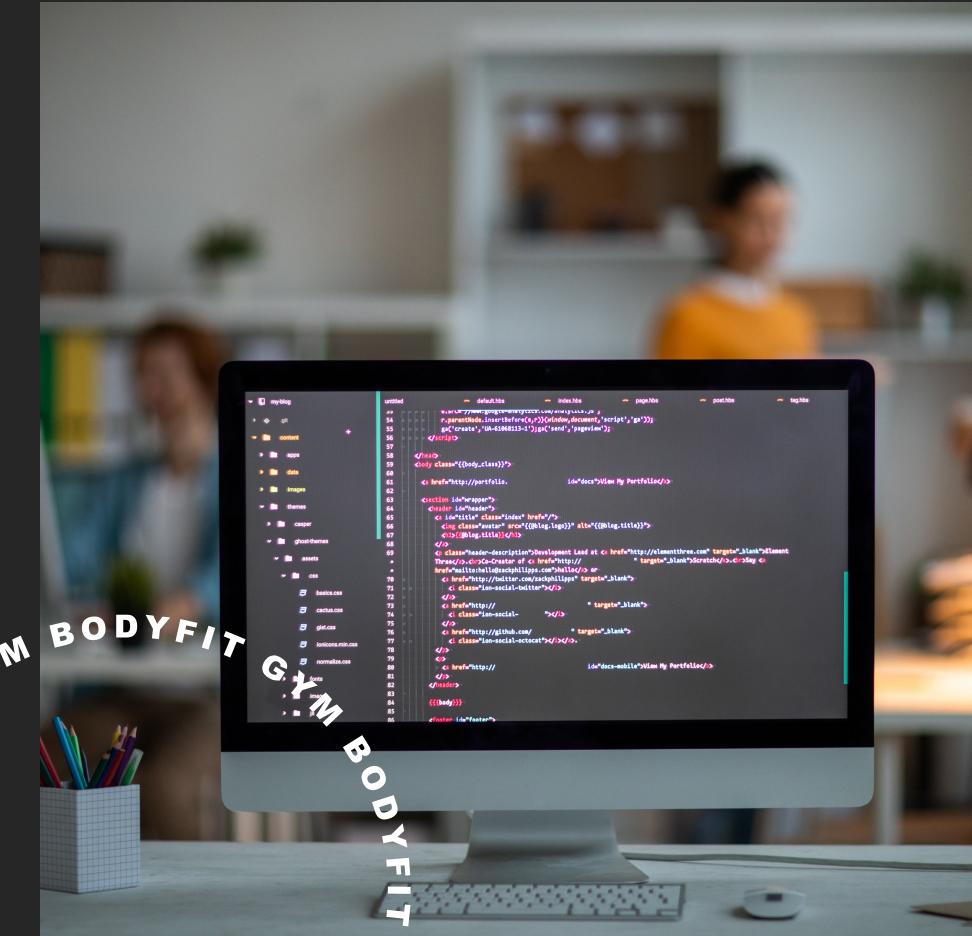
Point system
Menu selections (tty-prompt)
Case/When
If/elsif/else x 500 (Sorry!)
Methods
Class

{Something about error handling and testing}

GEMS

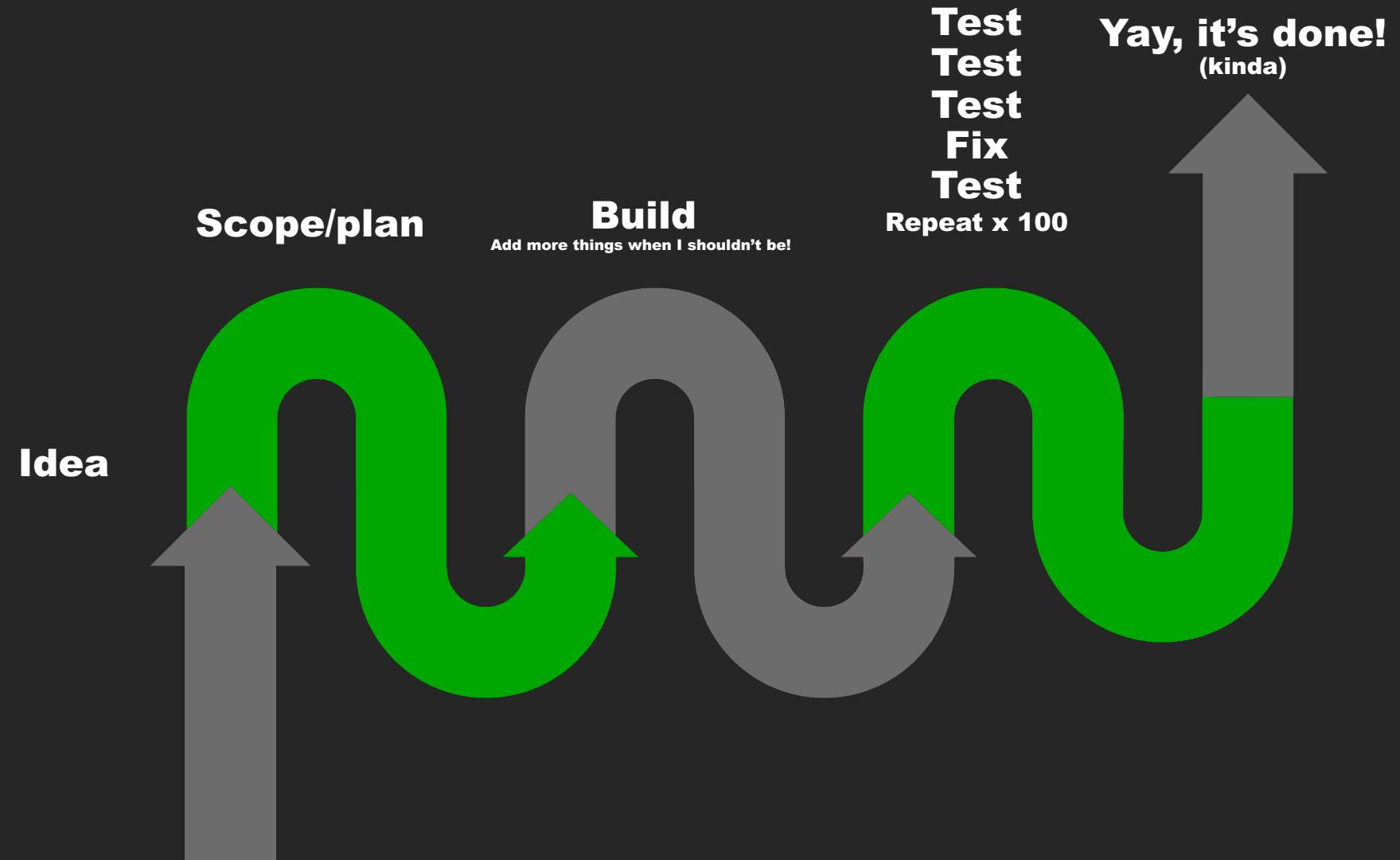
Tty-prompt
Tty-scroll
Fileutils
Down
Colourize

BODYFIT GYM BODYFIT GYM BODYFIT GYM BODYFIT GYM



```
united      - defaultable   - indexable   - paginable   - positive   - tagline
1  var ws = require('ws');
2  var express = require('express');
3  var http = require('http');
4  var app = express();
5  var server = http.createServer(app);
6  var io = require('socket.io')(server);
7  var socket;
8  var blog;
9  var posts;
10
11  // Create a new post
12  app.get('/create', function(req, res) {
13    res.sendFile(__dirname + '/views/post.html');
14  });
15
16  // Create a new post
17  app.post('/create', function(req, res) {
18    var post = req.body;
19    var id = Date.now();
20    var title = post.title;
21    var body = post.body;
22    var date = post.date;
23    var author = post.author;
24    var category = post.category;
25
26    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
27
28    res.redirect('/');
29  });
30
31
32  // Get all posts
33  app.get('/', function(req, res) {
34    res.send(blog.posts);
35  });
36
37
38  // Get a single post
39  app.get('/post/:id', function(req, res) {
40    var id = req.params.id;
41    var post = blog.posts[id];
42
43    if (!post) {
44      res.status(404).end();
45      return;
46    }
47
48    res.send(post);
49  });
50
51
52  // Delete a post
53  app.delete('/post/:id', function(req, res) {
54    var id = req.params.id;
55
56    if (!blog.posts[id]) {
57      res.status(404).end();
58      return;
59    }
60
61    delete blog.posts[id];
62
63    res.end();
64  });
65
66
67  // Create a new post
68  app.get('/create', function(req, res) {
69    res.sendFile(__dirname + '/views/post.html');
70  });
71
72
73  // Create a new post
74  app.post('/create', function(req, res) {
75    var post = req.body;
76    var id = Date.now();
77    var title = post.title;
78    var body = post.body;
79    var date = post.date;
80    var author = post.author;
81    var category = post.category;
82
83    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
84
85    res.redirect('/');
86  });
87
88
89  // Get all posts
90  app.get('/', function(req, res) {
91    res.send(blog.posts);
92  });
93
94
95  // Get a single post
96  app.get('/post/:id', function(req, res) {
97    var id = req.params.id;
98    var post = blog.posts[id];
99
100   if (!post) {
101     res.status(404).end();
102     return;
103   }
104
105   res.send(post);
106 });
107
108
109  // Delete a post
110  app.delete('/post/:id', function(req, res) {
111    var id = req.params.id;
112
113    if (!blog.posts[id]) {
114      res.status(404).end();
115      return;
116    }
117
118    delete blog.posts[id];
119
120    res.end();
121  });
122
123
124  // Create a new post
125  app.get('/create', function(req, res) {
126    res.sendFile(__dirname + '/views/post.html');
127  });
128
129
130  // Create a new post
131  app.post('/create', function(req, res) {
132    var post = req.body;
133    var id = Date.now();
134    var title = post.title;
135    var body = post.body;
136    var date = post.date;
137    var author = post.author;
138    var category = post.category;
139
140    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
141
142    res.redirect('/');
143  });
144
145
146  // Get all posts
147  app.get('/', function(req, res) {
148    res.send(blog.posts);
149  });
150
151
152  // Get a single post
153  app.get('/post/:id', function(req, res) {
154    var id = req.params.id;
155    var post = blog.posts[id];
156
157    if (!post) {
158      res.status(404).end();
159      return;
160    }
161
162    res.send(post);
163  });
164
165
166  // Delete a post
167  app.delete('/post/:id', function(req, res) {
168    var id = req.params.id;
169
170    if (!blog.posts[id]) {
171      res.status(404).end();
172      return;
173    }
174
175    delete blog.posts[id];
176
177    res.end();
178  });
179
180
181  // Create a new post
182  app.get('/create', function(req, res) {
183    res.sendFile(__dirname + '/views/post.html');
184  });
185
186
187  // Create a new post
188  app.post('/create', function(req, res) {
189    var post = req.body;
190    var id = Date.now();
191    var title = post.title;
192    var body = post.body;
193    var date = post.date;
194    var author = post.author;
195    var category = post.category;
196
197    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
198
199    res.redirect('/');
200  });
201
202
203  // Get all posts
204  app.get('/', function(req, res) {
205    res.send(blog.posts);
206  });
207
208
209  // Get a single post
210  app.get('/post/:id', function(req, res) {
211    var id = req.params.id;
212    var post = blog.posts[id];
213
214    if (!post) {
215      res.status(404).end();
216      return;
217    }
218
219    res.send(post);
220  });
221
222
223  // Delete a post
224  app.delete('/post/:id', function(req, res) {
225    var id = req.params.id;
226
227    if (!blog.posts[id]) {
228      res.status(404).end();
229      return;
230    }
231
232    delete blog.posts[id];
233
234    res.end();
235  });
236
237
238  // Create a new post
239  app.get('/create', function(req, res) {
240    res.sendFile(__dirname + '/views/post.html');
241  });
242
243
244  // Create a new post
245  app.post('/create', function(req, res) {
246    var post = req.body;
247    var id = Date.now();
248    var title = post.title;
249    var body = post.body;
250    var date = post.date;
251    var author = post.author;
252    var category = post.category;
253
254    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
255
256    res.redirect('/');
257  });
258
259
260  // Get all posts
261  app.get('/', function(req, res) {
262    res.send(blog.posts);
263  });
264
265
266  // Get a single post
267  app.get('/post/:id', function(req, res) {
268    var id = req.params.id;
269    var post = blog.posts[id];
270
271    if (!post) {
272      res.status(404).end();
273      return;
274    }
275
276    res.send(post);
277  });
278
279
280  // Delete a post
281  app.delete('/post/:id', function(req, res) {
282    var id = req.params.id;
283
284    if (!blog.posts[id]) {
285      res.status(404).end();
286      return;
287    }
288
289    delete blog.posts[id];
290
291    res.end();
292  });
293
294
295  // Create a new post
296  app.get('/create', function(req, res) {
297    res.sendFile(__dirname + '/views/post.html');
298  });
299
300
301  // Create a new post
302  app.post('/create', function(req, res) {
303    var post = req.body;
304    var id = Date.now();
305    var title = post.title;
306    var body = post.body;
307    var date = post.date;
308    var author = post.author;
309    var category = post.category;
310
311    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
312
313    res.redirect('/');
314  });
315
316
317  // Get all posts
318  app.get('/', function(req, res) {
319    res.send(blog.posts);
320  });
321
322
323  // Get a single post
324  app.get('/post/:id', function(req, res) {
325    var id = req.params.id;
326    var post = blog.posts[id];
327
328    if (!post) {
329      res.status(404).end();
330      return;
331    }
332
333    res.send(post);
334  });
335
336
337  // Delete a post
338  app.delete('/post/:id', function(req, res) {
339    var id = req.params.id;
340
341    if (!blog.posts[id]) {
342      res.status(404).end();
343      return;
344    }
345
346    delete blog.posts[id];
347
348    res.end();
349  });
350
351
352  // Create a new post
353  app.get('/create', function(req, res) {
354    res.sendFile(__dirname + '/views/post.html');
355  });
356
357
358  // Create a new post
359  app.post('/create', function(req, res) {
360    var post = req.body;
361    var id = Date.now();
362    var title = post.title;
363    var body = post.body;
364    var date = post.date;
365    var author = post.author;
366    var category = post.category;
367
368    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
369
370    res.redirect('/');
371  });
372
373
374  // Get all posts
375  app.get('/', function(req, res) {
376    res.send(blog.posts);
377  });
378
379
380  // Get a single post
381  app.get('/post/:id', function(req, res) {
382    var id = req.params.id;
383    var post = blog.posts[id];
384
385    if (!post) {
386      res.status(404).end();
387      return;
388    }
389
390    res.send(post);
391  });
392
393
394  // Delete a post
395  app.delete('/post/:id', function(req, res) {
396    var id = req.params.id;
397
398    if (!blog.posts[id]) {
399      res.status(404).end();
400      return;
401    }
402
403    delete blog.posts[id];
404
405    res.end();
406  });
407
408
409  // Create a new post
410  app.get('/create', function(req, res) {
411    res.sendFile(__dirname + '/views/post.html');
412  });
413
414
415  // Create a new post
416  app.post('/create', function(req, res) {
417    var post = req.body;
418    var id = Date.now();
419    var title = post.title;
420    var body = post.body;
421    var date = post.date;
422    var author = post.author;
423    var category = post.category;
424
425    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
426
427    res.redirect('/');
428  });
429
430
431  // Get all posts
432  app.get('/', function(req, res) {
433    res.send(blog.posts);
434  });
435
436
437  // Get a single post
438  app.get('/post/:id', function(req, res) {
439    var id = req.params.id;
440    var post = blog.posts[id];
441
442    if (!post) {
443      res.status(404).end();
444      return;
445    }
446
447    res.send(post);
448  });
449
450
451  // Delete a post
452  app.delete('/post/:id', function(req, res) {
453    var id = req.params.id;
454
455    if (!blog.posts[id]) {
456      res.status(404).end();
457      return;
458    }
459
460    delete blog.posts[id];
461
462    res.end();
463  });
464
465
466  // Create a new post
467  app.get('/create', function(req, res) {
468    res.sendFile(__dirname + '/views/post.html');
469  });
470
471
472  // Create a new post
473  app.post('/create', function(req, res) {
474    var post = req.body;
475    var id = Date.now();
476    var title = post.title;
477    var body = post.body;
478    var date = post.date;
479    var author = post.author;
480    var category = post.category;
481
482    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
483
484    res.redirect('/');
485  });
486
487
488  // Get all posts
489  app.get('/', function(req, res) {
490    res.send(blog.posts);
491  });
492
493
494  // Get a single post
495  app.get('/post/:id', function(req, res) {
496    var id = req.params.id;
497    var post = blog.posts[id];
498
499    if (!post) {
500      res.status(404).end();
501      return;
502    }
503
504    res.send(post);
505  });
506
507
508  // Delete a post
509  app.delete('/post/:id', function(req, res) {
510    var id = req.params.id;
511
512    if (!blog.posts[id]) {
513      res.status(404).end();
514      return;
515    }
516
517    delete blog.posts[id];
518
519    res.end();
520  });
521
522
523  // Create a new post
524  app.get('/create', function(req, res) {
525    res.sendFile(__dirname + '/views/post.html');
526  });
527
528
529  // Create a new post
530  app.post('/create', function(req, res) {
531    var post = req.body;
532    var id = Date.now();
533    var title = post.title;
534    var body = post.body;
535    var date = post.date;
536    var author = post.author;
537    var category = post.category;
538
539    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
540
541    res.redirect('/');
542  });
543
544
545  // Get all posts
546  app.get('/', function(req, res) {
547    res.send(blog.posts);
548  });
549
550
551  // Get a single post
552  app.get('/post/:id', function(req, res) {
553    var id = req.params.id;
554    var post = blog.posts[id];
555
556    if (!post) {
557      res.status(404).end();
558      return;
559    }
560
561    res.send(post);
562  });
563
564
565  // Delete a post
566  app.delete('/post/:id', function(req, res) {
567    var id = req.params.id;
568
569    if (!blog.posts[id]) {
570      res.status(404).end();
571      return;
572    }
573
574    delete blog.posts[id];
575
576    res.end();
577  });
578
579
580  // Create a new post
581  app.get('/create', function(req, res) {
582    res.sendFile(__dirname + '/views/post.html');
583  });
584
585
586  // Create a new post
587  app.post('/create', function(req, res) {
588    var post = req.body;
589    var id = Date.now();
590    var title = post.title;
591    var body = post.body;
592    var date = post.date;
593    var author = post.author;
594    var category = post.category;
595
596    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
597
598    res.redirect('/');
599  });
600
601
602  // Get all posts
603  app.get('/', function(req, res) {
604    res.send(blog.posts);
605  });
606
607
608  // Get a single post
609  app.get('/post/:id', function(req, res) {
610    var id = req.params.id;
611    var post = blog.posts[id];
612
613    if (!post) {
614      res.status(404).end();
615      return;
616    }
617
618    res.send(post);
619  });
620
621
622  // Delete a post
623  app.delete('/post/:id', function(req, res) {
624    var id = req.params.id;
625
626    if (!blog.posts[id]) {
627      res.status(404).end();
628      return;
629    }
630
631    delete blog.posts[id];
632
633    res.end();
634  });
635
636
637  // Create a new post
638  app.get('/create', function(req, res) {
639    res.sendFile(__dirname + '/views/post.html');
640  });
641
642
643  // Create a new post
644  app.post('/create', function(req, res) {
645    var post = req.body;
646    var id = Date.now();
647    var title = post.title;
648    var body = post.body;
649    var date = post.date;
650    var author = post.author;
651    var category = post.category;
652
653    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
654
655    res.redirect('/');
656  });
657
658
659  // Get all posts
660  app.get('/', function(req, res) {
661    res.send(blog.posts);
662  });
663
664
665  // Get a single post
666  app.get('/post/:id', function(req, res) {
667    var id = req.params.id;
668    var post = blog.posts[id];
669
670    if (!post) {
671      res.status(404).end();
672      return;
673    }
674
675    res.send(post);
676  });
677
678
679  // Delete a post
680  app.delete('/post/:id', function(req, res) {
681    var id = req.params.id;
682
683    if (!blog.posts[id]) {
684      res.status(404).end();
685      return;
686    }
687
688    delete blog.posts[id];
689
690    res.end();
691  });
692
693
694  // Create a new post
695  app.get('/create', function(req, res) {
696    res.sendFile(__dirname + '/views/post.html');
697  });
698
699
700  // Create a new post
701  app.post('/create', function(req, res) {
702    var post = req.body;
703    var id = Date.now();
704    var title = post.title;
705    var body = post.body;
706    var date = post.date;
707    var author = post.author;
708    var category = post.category;
709
710    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
711
712    res.redirect('/');
713  });
714
715
716  // Get all posts
717  app.get('/', function(req, res) {
718    res.send(blog.posts);
719  });
720
721
722  // Get a single post
723  app.get('/post/:id', function(req, res) {
724    var id = req.params.id;
725    var post = blog.posts[id];
726
727    if (!post) {
728      res.status(404).end();
729      return;
730    }
731
732    res.send(post);
733  });
734
735
736  // Delete a post
737  app.delete('/post/:id', function(req, res) {
738    var id = req.params.id;
739
740    if (!blog.posts[id]) {
741      res.status(404).end();
742      return;
743    }
744
745    delete blog.posts[id];
746
747    res.end();
748  });
749
750
751  // Create a new post
752  app.get('/create', function(req, res) {
753    res.sendFile(__dirname + '/views/post.html');
754  });
755
756
757  // Create a new post
758  app.post('/create', function(req, res) {
759    var post = req.body;
760    var id = Date.now();
761    var title = post.title;
762    var body = post.body;
763    var date = post.date;
764    var author = post.author;
765    var category = post.category;
766
767    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
768
769    res.redirect('/');
770  });
771
772
773  // Get all posts
774  app.get('/', function(req, res) {
775    res.send(blog.posts);
776  });
777
778
779  // Get a single post
780  app.get('/post/:id', function(req, res) {
781    var id = req.params.id;
782    var post = blog.posts[id];
783
784    if (!post) {
785      res.status(404).end();
786      return;
787    }
788
789    res.send(post);
790  });
791
792
793  // Delete a post
794  app.delete('/post/:id', function(req, res) {
795    var id = req.params.id;
796
797    if (!blog.posts[id]) {
798      res.status(404).end();
799      return;
800    }
801
802    delete blog.posts[id];
803
804    res.end();
805  });
806
807
808  // Create a new post
809  app.get('/create', function(req, res) {
810    res.sendFile(__dirname + '/views/post.html');
811  });
812
813
814  // Create a new post
815  app.post('/create', function(req, res) {
816    var post = req.body;
817    var id = Date.now();
818    var title = post.title;
819    var body = post.body;
820    var date = post.date;
821    var author = post.author;
822    var category = post.category;
823
824    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
825
826    res.redirect('/');
827  });
828
829
830  // Get all posts
831  app.get('/', function(req, res) {
832    res.send(blog.posts);
833  });
834
835
836  // Get a single post
837  app.get('/post/:id', function(req, res) {
838    var id = req.params.id;
839    var post = blog.posts[id];
840
841    if (!post) {
842      res.status(404).end();
843      return;
844    }
845
846    res.send(post);
847  });
848
849
850  // Delete a post
851  app.delete('/post/:id', function(req, res) {
852    var id = req.params.id;
853
854    if (!blog.posts[id]) {
855      res.status(404).end();
856      return;
857    }
858
859    delete blog.posts[id];
860
861    res.end();
862  });
863
864
865  // Create a new post
866  app.get('/create', function(req, res) {
867    res.sendFile(__dirname + '/views/post.html');
868  });
869
870
871  // Create a new post
872  app.post('/create', function(req, res) {
873    var post = req.body;
874    var id = Date.now();
875    var title = post.title;
876    var body = post.body;
877    var date = post.date;
878    var author = post.author;
879    var category = post.category;
880
881    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
882
883    res.redirect('/');
884  });
885
886
887  // Get all posts
888  app.get('/', function(req, res) {
889    res.send(blog.posts);
890  });
891
892
893  // Get a single post
894  app.get('/post/:id', function(req, res) {
895    var id = req.params.id;
896    var post = blog.posts[id];
897
898    if (!post) {
899      res.status(404).end();
900      return;
901    }
902
903    res.send(post);
904  });
905
906
907  // Delete a post
908  app.delete('/post/:id', function(req, res) {
909    var id = req.params.id;
910
911    if (!blog.posts[id]) {
912      res.status(404).end();
913      return;
914    }
915
916    delete blog.posts[id];
917
918    res.end();
919  });
920
921
922  // Create a new post
923  app.get('/create', function(req, res) {
924    res.sendFile(__dirname + '/views/post.html');
925  });
926
927
928  // Create a new post
929  app.post('/create', function(req, res) {
930    var post = req.body;
931    var id = Date.now();
932    var title = post.title;
933    var body = post.body;
934    var date = post.date;
935    var author = post.author;
936    var category = post.category;
937
938    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
939
940    res.redirect('/');
941  });
942
943
944  // Get all posts
945  app.get('/', function(req, res) {
946    res.send(blog.posts);
947  });
948
949
950  // Get a single post
951  app.get('/post/:id', function(req, res) {
952    var id = req.params.id;
953    var post = blog.posts[id];
954
955    if (!post) {
956      res.status(404).end();
957      return;
958    }
959
960    res.send(post);
961  });
962
963
964  // Delete a post
965  app.delete('/post/:id', function(req, res) {
966    var id = req.params.id;
967
968    if (!blog.posts[id]) {
969      res.status(404).end();
970      return;
971    }
972
973    delete blog.posts[id];
974
975    res.end();
976  });
977
978
979  // Create a new post
980  app.get('/create', function(req, res) {
981    res.sendFile(__dirname + '/views/post.html');
982  });
983
984
985  // Create a new post
986  app.post('/create', function(req, res) {
987    var post = req.body;
988    var id = Date.now();
989    var title = post.title;
990    var body = post.body;
991    var date = post.date;
992    var author = post.author;
993    var category = post.category;
994
995    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
996
997    res.redirect('/');
998  });
999
1000
1001  // Get all posts
1002  app.get('/', function(req, res) {
1003    res.send(blog.posts);
1004  });
1005
1006
1007  // Get a single post
1008  app.get('/post/:id', function(req, res) {
1009    var id = req.params.id;
1010    var post = blog.posts[id];
1011
1012    if (!post) {
1013      res.status(404).end();
1014      return;
1015    }
1016
1017    res.send(post);
1018  });
1019
1020
1021  // Delete a post
1022  app.delete('/post/:id', function(req, res) {
1023    var id = req.params.id;
1024
1025    if (!blog.posts[id]) {
1026      res.status(404).end();
1027      return;
1028    }
1029
1030    delete blog.posts[id];
1031
1032    res.end();
1033  });
1034
1035
1036  // Create a new post
1037  app.get('/create', function(req, res) {
1038    res.sendFile(__dirname + '/views/post.html');
1039  });
1040
1041
1042  // Create a new post
1043  app.post('/create', function(req, res) {
1044    var post = req.body;
1045    var id = Date.now();
1046    var title = post.title;
1047    var body = post.body;
1048    var date = post.date;
1049    var author = post.author;
1050    var category = post.category;
1051
1052    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
1053
1054    res.redirect('/');
1055  });
1056
1057
1058  // Get all posts
1059  app.get('/', function(req, res) {
1060    res.send(blog.posts);
1061  });
1062
1063
1064  // Get a single post
1065  app.get('/post/:id', function(req, res) {
1066    var id = req.params.id;
1067    var post = blog.posts[id];
1068
1069    if (!post) {
1070      res.status(404).end();
1071      return;
1072    }
1073
1074    res.send(post);
1075  });
1076
1077
1078  // Delete a post
1079  app.delete('/post/:id', function(req, res) {
1080    var id = req.params.id;
1081
1082    if (!blog.posts[id]) {
1083      res.status(404).end();
1084      return;
1085    }
1086
1087    delete blog.posts[id];
1088
1089    res.end();
1090  });
1091
1092
1093  // Create a new post
1094  app.get('/create', function(req, res) {
1095    res.sendFile(__dirname + '/views/post.html');
1096  });
1097
1098
1099  // Create a new post
1100  app.post('/create', function(req, res) {
1101    var post = req.body;
1102    var id = Date.now();
1103    var title = post.title;
1104    var body = post.body;
1105    var date = post.date;
1106    var author = post.author;
1107    var category = post.category;
1108
1109    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
1110
1111    res.redirect('/');
1112  });
1113
1114
1115  // Get all posts
1116  app.get('/', function(req, res) {
1117    res.send(blog.posts);
1118  });
1119
1120
1121  // Get a single post
1122  app.get('/post/:id', function(req, res) {
1123    var id = req.params.id;
1124    var post = blog.posts[id];
1125
1126    if (!post) {
1127      res.status(404).end();
1128      return;
1129    }
1130
1131    res.send(post);
1132  });
1133
1134
1135  // Delete a post
1136  app.delete('/post/:id', function(req, res) {
1137    var id = req.params.id;
1138
1139    if (!blog.posts[id]) {
1140      res.status(404).end();
1141      return;
1142    }
1143
1144    delete blog.posts[id];
1145
1146    res.end();
1147  });
1148
1149
1150  // Create a new post
1151  app.get('/create', function(req, res) {
1152    res.sendFile(__dirname + '/views/post.html');
1153  });
1154
1155
1156  // Create a new post
1157  app.post('/create', function(req, res) {
1158    var post = req.body;
1159    var id = Date.now();
1160    var title = post.title;
1161    var body = post.body;
1162    var date = post.date;
1163    var author = post.author;
1164    var category = post.category;
1165
1166    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
1167
1168    res.redirect('/');
1169  });
1170
1171
1172  // Get all posts
1173  app.get('/', function(req, res) {
1174    res.send(blog.posts);
1175  });
1176
1177
1178  // Get a single post
1179  app.get('/post/:id', function(req, res) {
1180    var id = req.params.id;
1181    var post = blog.posts[id];
1182
1183    if (!post) {
1184      res.status(404).end();
1185      return;
1186    }
1187
1188    res.send(post);
1189  });
1190
1191
1192  // Delete a post
1193  app.delete('/post/:id', function(req, res) {
1194    var id = req.params.id;
1195
1196    if (!blog.posts[id]) {
1197      res.status(404).end();
1198      return;
1199    }
1200
1201    delete blog.posts[id];
1202
1203    res.end();
1204  });
1205
1206
1207  // Create a new post
1208  app.get('/create', function(req, res) {
1209    res.sendFile(__dirname + '/views/post.html');
1210  });
1211
1212
1213  // Create a new post
1214  app.post('/create', function(req, res) {
1215    var post = req.body;
1216    var id = Date.now();
1217    var title = post.title;
1218    var body = post.body;
1219    var date = post.date;
1220    var author = post.author;
1221    var category = post.category;
1222
1223    blog.posts[id] = {id: id, title: title, body: body, date: date, author: author, category: category};
1224
1225    res.redirect('/');
1226  });
1227
1228
1229  // Get all posts
1230  app.get('/', function(req, res) {
1231    res.send(blog.posts);
1232  });
1233
1234
1235  // Get a single post
1236  app.get('/post/:id', function(req, res) {
1237    var id = req.params.id;
1238    var post = blog.posts[id];
1239
1240    if (!post) {
1241      res.status(404).end();
1242      return;
1243    }
1244
1245    res.send(post);
1246  });
1247
1248
1249  // Delete a post
1250  app.delete('/post/:id', function(req, res) {
1251    var id
```

DEV PROCESS



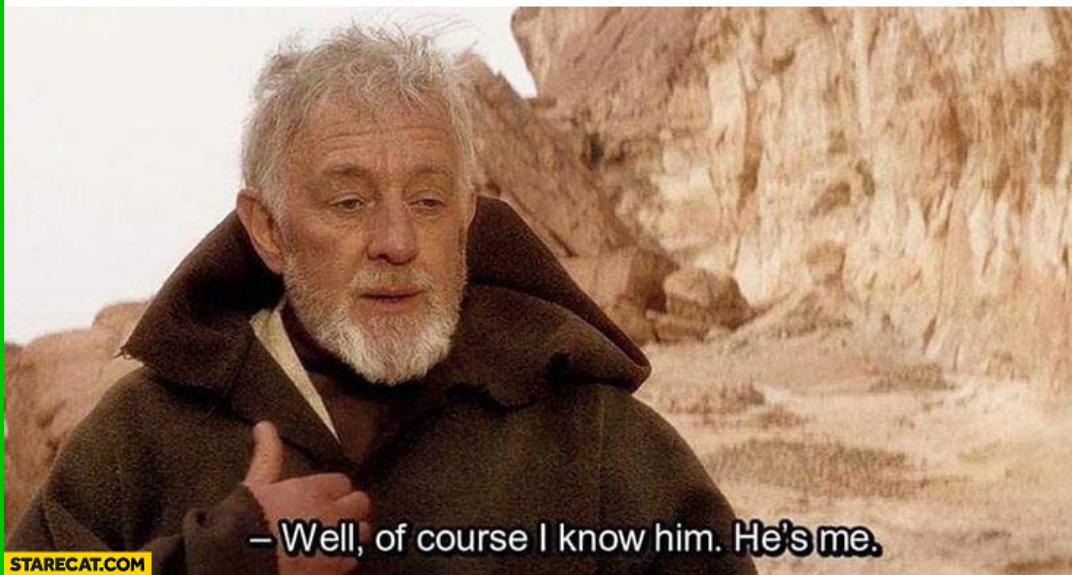
BODY
BODY
SHAPE YOUR **BODY**

Point system

```
src > Index.rb
246     fitness_level = 18
247
248 #If they are male overweight and have a low fitness level
249 elsif b == 7 && f == 1
250     fitness_level = 19
251 #If they are male overweight and have a medium fitness level
252 elsif b == 7 && f == 2
253     fitness_level = 20
254 #If they are male overweight and have a high fitness level
255 elsif b == 7 && f == 3
256     fitness_level = 21
257
258 #If they are male obese and have a low fitness level
259 elsif b == 8 && f == 1
260     fitness_level = 22
261 #If they are male obese and have a medium fitness level
262 elsif b == 8 && f == 2
263     fitness_level = 23
264 #If they are male obese and have a high fitness level
265 else b == 8 && f == 3
266     fitness_level = 24
267 end
268
269
270 #Final output/outcome - Calculates a solution based of the answers given.
271
272 # For real program, the download link would be different per else if - for assessment purposes, they are only different per goal selected, rather than
having 70+ different documents.
273
274
275 #If user selected loose weight - Refer to conditional.rb for fitness_level number system
276 if goal == 1 && fitness_level == 1
277     puts "You want to loose weight, but you are not a vey active person. It seems you also don't have much weight to loose. Perhaps you should seek
professional support for your goals. I have added a list of support lines "
278     tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
279     FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
280
281 elsif goal == 1 && fitness_level == 2
282     puts "Second fitness option, your file will download soon"
283     tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
284     FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
285
286 elsif goal == 1 && fitness_level == 3
287     puts "Second fitness option, your file will download soon"
288     tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
289     FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
290 elsif goal == 1 && fitness_level == 4
291     puts "Second fitness option, your file will download soon"
292     tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
293     FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
294 elsif goal == 1 && fitness_level == 5
295     puts "Second fitness option, your file will download soon"
296     tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
297     FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
298 elsif goal == 1 && fitness_level == 6
299     puts "Second fitness option, your file will download soon"
```

I AM SORRY ASSESSOR!

When you read some incredibly bad code, thinking "What moron wrote this...", but halfway through it starts to become familiar.



I DON'T KNOW WHO WROTE THIS CODE



BUT I WILL FIND YOU AND I WILL KILL YOU

.memegenerator.net

BODY BODY BODY

SHAPE YOUR BODY

FIT FIT FIT

Menu select & Case/When

```
DOME.md M Index.rb M X conditional.rb mail.rb M testing.rb artmessages.rb M  
Index.rb  
  
#sleep (1)  
  
testing = prompt.select("To get started, we are going to need to get some details from yourself. We will look at the goals you want to achieve provide you with a custom plan that is just right for you! First we need to know what your goal is, do you want to Loose weight, Build muscle or Increase energy?") do | menu|  
  menu.choice 'loose weight'  
  menu.choice 'build muscle'  
  menu.choice 'increase energy'  
end  
  
case  
when testing == "loose weight"  
  puts "It's great to focus on loosing weight! But remember, when we start becoming healthier and excercise we also naturally gain muscle, when starting your journey, take lots of pictures of your progress rather than focusing on the scales. Don't worry, your plan will still custom to your weight loss goal!"  
  goal_type = "Loose Weight"  
  goal = 1  
  puts "loose"  
  
when testing == "build muscle"  
  puts "You are my people! Building muscle not only makes you feel good, it also means you look good! The more muscle you have, the more your body breaks down fat. We will make sure your custom plan has lots of weights to keep you excited!"  
  goal_type = "Build Muscle"  
  goal = 2  
  puts "build"  
  
when testing == "increase energy"  
  puts "Uh! I feel you! There is nothing worse than being tired. How great is it that we can do something about this by filling out body with the energy and nutrients it needs to feel better. If this is your goal, it's important to also go get your bloods tested to ensure you are Iron deficent. Iron deficiency is the most common nutritional disorder affecting about 20-25% of the world's population. It's a real drain Trust me!"  
  goal_type = "Increase Energy"  
  goal = 3  
  puts "increase"  
end  
  
#system "clear"  
  
puts "Thanks for that! Now we are going to calculate your Body Mass Index, this is calculated off your height and weight"  
  
puts "It is useful to consider BMI alongside waist circumference, as waist measurement helps to assess risk by measuring the amount of fat carried around your middle."  
  
BMI is a useful measurement for most people over 18 years old. But it is only an estimate and it doesn't take into account age, ethnicity, gender and body composition. Like we mentioned earlier, don't get too focused on BMI ro the scales when starting and take lots of before and afters, this just helps us create a custom plan for you! We will also take into consideration your current excercise levels."
```



ASCII ART

Selection View Go Run Terminal Window Help

artmessages.rb — MatildaMorton_T1A3

... README.md Index.rb artmessages.rb X

src > artmessages.rb

```
1
2 def logo
3   puts "
4
5
6
7
8
9
10
11
12
13
14
15   ".colorize(:blue)
16 end
17
18
19 def awesome
20
21   puts "
22
23
24
25
26
27
28
29
30   ".colorize(:magenta)
31 end
32
33
34
35 def fittitle
36   puts "
37
38
39
40
41
42
43
44
45
46
47
48
49   ".colorize(:magenta)
```

testing Aa ab * No results

RODPI

AWEsome WORD

HOUt iT ROD M

**IT'S TIME TO
SHAPE YOUR BODY**

Image download

```
DME.md | Index.rb | final.rb | final.rb
> testing
Aa ab * No results ↑ ↓ ⌂
if goal == 1 && fitness_level == 1
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")

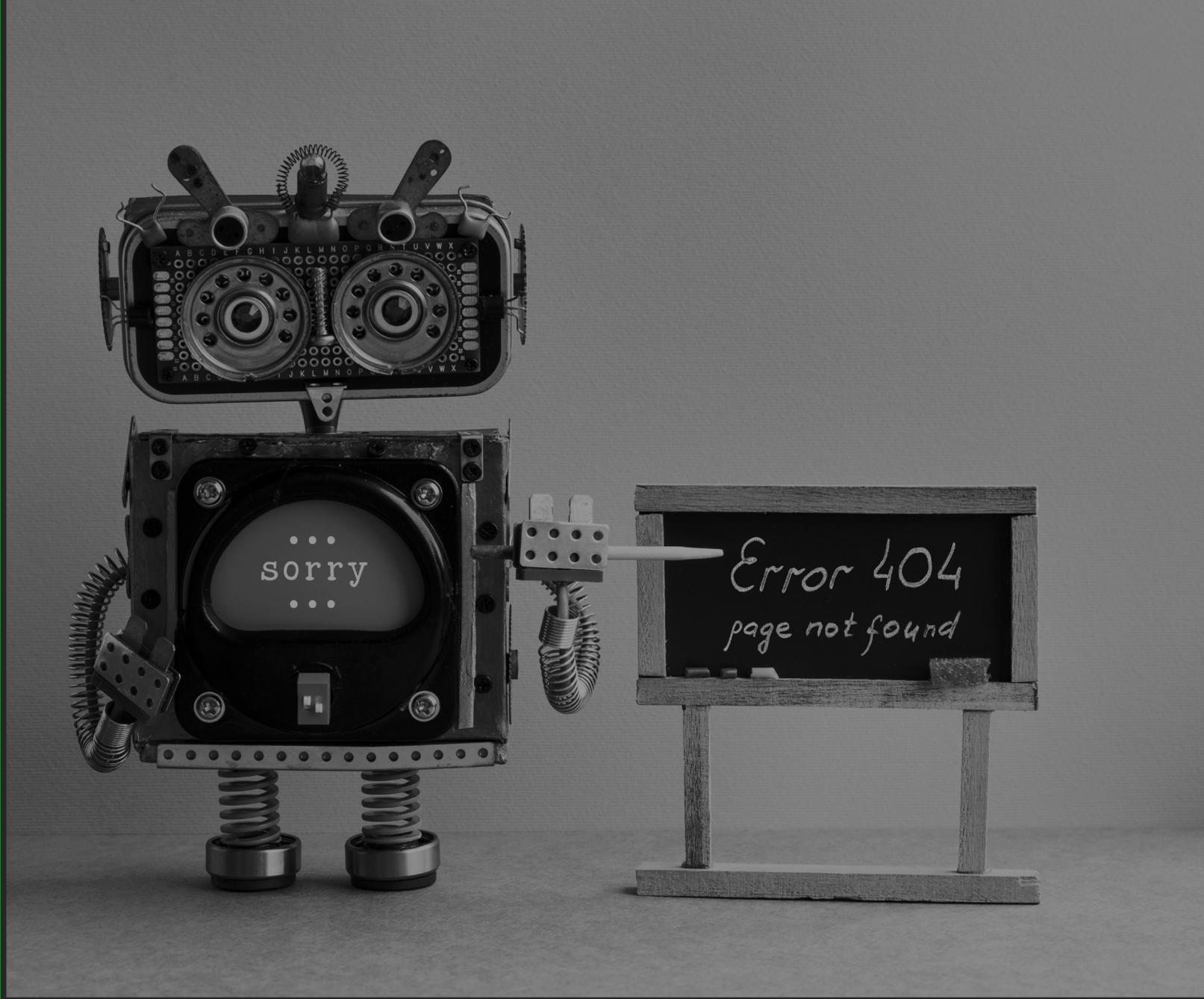
elsif goal == 1 && fitness_level == 2
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")

elsif goal == 1 && fitness_level == 3
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 4
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 5
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 6
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 7
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 8
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 9
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 10
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 11
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
elsif goal == 1 && fitness_level == 12
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
else
  puts "Second fitness option, your file will download soon"
  tempfile = Down.download("https://github.com/MatildaMortonCode/MatildaMorton_Fitnessplan/blob/main/docs/Loose-weight.%20pdf.pdf")
  FileUtils.mv(tempfile.path, "./#{tempfile.original_filename}")
end
```

BODYFIT

SHAPE YOUR BODY

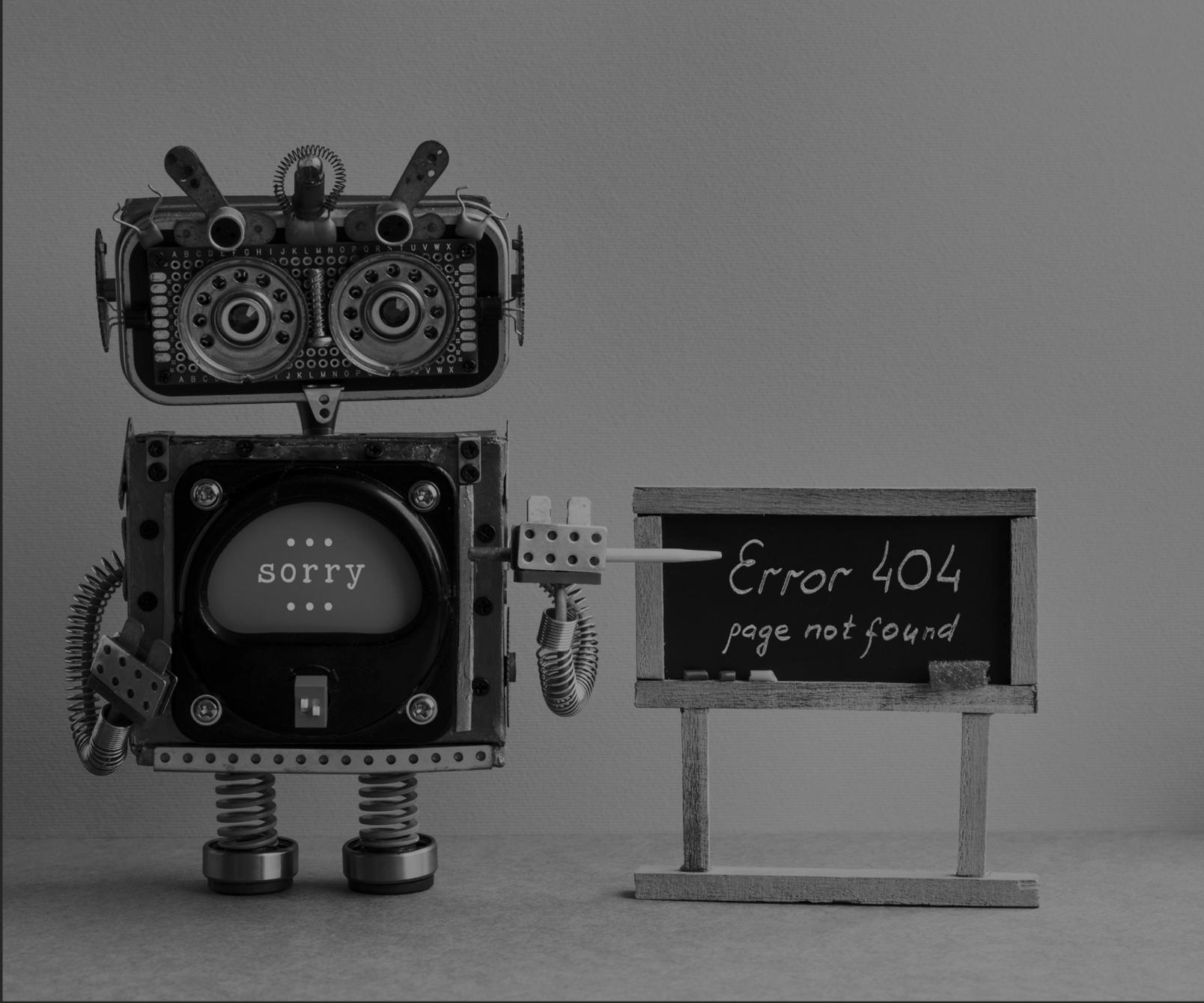
Testing



BODYFIT

SHAPE YOUR BODY

Scripting



Challenges

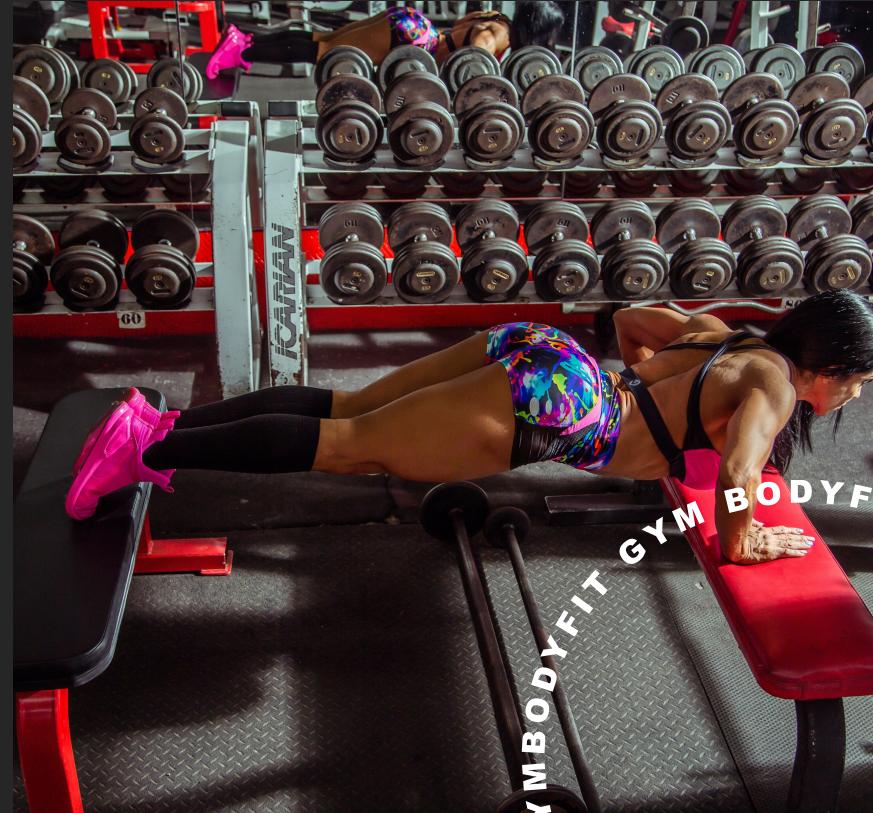
TAKE CARE OF YOUR BODY

Getting started!

Four way conditional - && x 4

Imagery -Rmagick

Scope of program



Favourite bits



BODYFIT

KNOWING I HAVE MADE THE RIGHT CHOICE

THE
BODYFIT
METHOD

BODYFIT



Ethics



LET'S LOOK AT IT LIVE!

THANKS
THANKS
THANKS
THANKS

HOPE YOU ENJOYED MY PRSSO!