

Best Of

Actualités les plus lues

Semaine
Mois
Année



JavaScript, Python et Java sont les langages de développement les plus utilisés par les développeurs de logiciel actif tandis que C# perd du terrain selon l'enquête State of the Developer Nation (https://programmation.developpez.com/actu/309874/JavaScript-Python-et-Java-sont-les-langages-de-developpement-les-plus-utilises-par-les-developpeurs-de-logiciel-actif-tandis-que-C-perd-du-terrain-selon-l-enquete-State-of-the-Developer-Nation/).

Python est désormais plus populaire que Java et occupe la deuxième place de l'index TIOBE en novembre, Java ne fait pas partie du duo de tête pour la première fois en 20 ans (https://programmation.developpez.com/actu/310194/Python-est-desormais-plus-populaire-que-Java-et-occupe-la-deuxieme-place-de-l-index-TIOBE-en-novembre-Java-ne-fait-pas-partie-du-duo-de-tete-pour-la-premiere-fois-en-20-ans/).

Global Knowledge lance Appdev, une expérience d'apprentissage mixte en développement d'application Web qui associe différents outils pédagogiques, comme le digital augmenté, un coding workshop, etc. (https://www.developpez.com/actu/310069/Global-Knowledge-lance-Appdev-une-experience-d-apprentissage-mixte-en-developpement-d-application-Web-qui-associe-differents-outils-pedagogiques-comme-le-digital-augmente-un-coding-workshop-etc/).

Node.js 15 est disponible et apporte le support de npm 7, une implémentation expérimentale AbortController ainsi qu'une mise à jour du moteur JavaScript V8 qui passe en version 8.6 (https://javascript.developpez.com/actu/309929/Node-js-15-est-disponible-et-apporte-le-support-de-npm-7-une-implementation-experimentale-AbortController-ainsi-qu-une-mise-a-jour-du-moteur-JavaScript-V8-qui-passe-en-version-8-6/).

Cours et tutoriels pour Débuter en JavaScript

- **Tutoriel d'introduction au langage JavaScript** (https://javascript.developpez.com/tutoriels/introduction/) par Serge P.
- **Cours d'introduction au langage JavaScript** (https://jacques-guizol.developpez.com/javascript/) par Jacques Guizol
- **Débutez en programmation avec le langage JavaScript** par Matthieu Deloison (https://programmation-facile.developpez.com/tutoriels/javascript/debutez-programmation-avec-langage-javascript/).

Forums JavaScript

- **Général JavaScript** (https://www.developpez.net/forums/f23/javascript/general-javascript/).
- **Bibliothèques & Frameworks JavaScript** (https://www.developpez.net/forums/f1187/javascript/bibliotheques-frameworks/).
- **Forum sur le framework JavaScript : NodeJS** (https://www.developpez.net/forums/f2137/javascript/nodejs/).
- **JavaScript - Forum sur le langage de programmation TypeScript** (https://www.developpez.net/forums/f2001/javascript/typescript/).
- **JavaScript - Forum sur le langage de programmation Dart** (https://www.developpez.net/forums/f1998/javascript/dart/).

Les meilleures ressources JavaScript

- **120 cours et tutoriels** (http://javascript.developpez.com/cours/) pour apprendre JavaScript
- La **FAQ JavaScript : 176 réponses** (http://javascript.developpez.com/faq/javascript/).
- La **FAQ jQuery**, bibliothèque JavaScript libre et multiplateforme : **46 réponses** (http://javascript.developpez.com/faq/jquery).
- La **FAQ sur le framework JavaScript MooTools : 67 réponses** (http://javascript.developpez.com/faq/mootools).
- La **FAQ sur la bibliothèque JavaScript ExtJS/Sencha : 59 réponses** (http://javascript.developpez.com/faq/extjs).
- La **FAQ sur l'API JavaScript Google Maps : 31 réponses** (http://javascript.developpez.com/faq/google-maps).
- **Téléchargez les sources JavaScript** (http://javascript.developpez.com/telecharger/index/categorie/251/Sources-JavaScript).
- **Livres JavaScript et critiques** de la rédaction (http://javascript.developpez.com/livres/).

Communauté JavaScript

- Découvrez une partie de l'équipe JavaScript (http://web.developpez.com/equipe/).
- Les billets blog forum JavaScript (http://www.developpez.net/forums/blog.php?blogcategoryid=44).
- Vous souhaitez rejoindre la rédaction ou proposer un tutoriel sur le langage de programmation JavaScript : contactez-nous par mail (http://javascript.developpez.com/email).



Exercices NodeJS pour apprendre le JavaScript côté serveur

Nombre d'auteurs : 5 - Nombre d'exercices : 8 - Dernière mise à jour : 23 avril 2016

Rechercher

Une sélection des meilleurs exercices, accessibles aux débutants, avec des énoncés clairs et complets suivis de solutions détaillées.

Grâce à l'entraide bénévole, les membres du club répondent à vos questions directement sur le forum et vous aident lors de l'apprentissage du langage.

Sommaire (/exercices/nodejs) ➤ Le langage JavaScript (?page=Le-langage-JavaScript) ➤ Les opérateurs conditionnels et le typage implicite (?page=Les-operateurs-conditionnels-et-le-typage-implicite).

- ➤ [Facile][Indispensable] Exercice 1.2.1 : Les structures conditionnelles avec l'instruction "if...else"
- ➤ [Facile][Indispensable] Exercice 1.2.2 : Les structures conditionnelles avec l'instruction "switch"
- ➤ [Intermédiaire][Indispensable] Exercice 1.2.3 : Les particularités du switch et l'utilisation de la bibliothèque standard
- ➤ [Facile][Indispensable] Exercice 1.2.4 : Le typage dynamique
- ➤ [Facile][Indispensable] Exercice 1.2.5 : L'opérateur conditionnel ternaire

◀ Le premier script (?page=Le-premier-script).

➤ [Facile][Indispensable] Exercice 1.2.1 : Les structures conditionnelles avec l'instruction "if...else"

Mis à jour le 3 mars 2016 par ABCIWEB (https://www.developpez.com/user/profil/388426/ABCIWEB), autran (https://www.developpez.com/user/profil/863457/autran), Gnuum (https://www.developpez.com/user/profil/153437/Gnuum), vermine (https://www.developpez.com/user/profil/226061/vermine).

Objectif

Manipuler les structures conditionnelles de type `if...else`.

Compétences

- Utilisation des structures conditionnelles de type `if...else`.
- Utilisation d'expressions booléennes.

Cours

Algèbre de Boole

L'algèbre de Boole est utilisée en programmation pour évaluer si une expression est vraie ou fausse.

Si on prend quelques exemples basiques :

VRAI et VRAI = VRAI

VRAI et FAUX = FAUX

VRAI ou FAUX = VRAI

Si on applique ces exemples au JavaScript cela donne :

Code javascript :

```
1 true && true === true
2 true && false === false
3 true || false === true
```

Sélectionner tout

Expression booléenne

Une expression booléenne en JavaScript est une expression qui s'évalue à **true** ou à **false**.

Cela peut, par exemple, être une égalité, une inégalité, une fonction retournant une valeur booléenne...

Code javascript :

```
1 var a = 0,
2     b = 1,
3     c = '1'
4 ;
5
6 console.log(a === 0); // Affiche true (égal à)
7 console.log(a === 1); // Affiche false
8 console.log(a === b); // Affiche false
9 console.log(a === 0 && b === 1); // Affiche true
10 console.log(a === 0 && b === 0); // Affiche false
11 console.log(a === 0 || b === 0); // Affiche true
12 console.log(b === c); // Affiche false (car le type est différent)
13 console.log(b == c); // Affiche true (car l'opérateur == essaie de faire fi du type si
14 e)
15 console.log(a < b); // Affiche true (inférieur à)
16 console.log(a >= b); // Affiche false (supérieur ou égal à)
17 console.log(a !== b); // Affiche true (différent de)
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

Sélectionner tout

Il est possible d'utiliser des parenthèses pour former des expressions imbriquées et forcer l'ordre de précedence (ordre d'évaluation).

Code javascript :

```
a === 0 && (b === 1 || b === 2)
```

Dans cet exemple, (b === 1 || b === 2) est évaluée en premier ; ce qui donne, si on détaille l'étape intermédiaire :

Code javascript :

```
a === 0 && true
```

Sélectionner tout

Sélectionner tout

Algorithmique et structures conditionnelles

Une des structures de base de l'algorithmique de la plupart des langages de programmation est le branchement conditionnel de type « si », « alors », « sinon ».

En algorithmique, la syntaxe pour employer ce branchement est :

Code :

```
1 si alors
2
3 sinon
4
```

Si on traduit en mots, cela consiste à :

- évaluer l'expression ;
- exécuter si l'évaluation donne VRAI ;

- ou exécuter dans le cas contraire.

L'instruction « sinon » n'est pas obligatoire, on peut écrire :

Code :

```
1 si alors
2
```

Sélectionner tout

Dans ce cas, on n'exécutera que si s'évalue à VRAI.

Structures conditionnelles en JavaScript

En JavaScript, les mots clés utilisés pour décrire une structure conditionnelle de ce type sont **if** et **else**:

Code javascript :

```
1 var a = 0;
2
3 if (a === 0) {
4     console.log('Je serai affiché !');
5 } else {
6     console.log('Je ne serai pas affiché...');
7 }
```

Sélectionner tout

Il est bien sûr possible d'imbriquer ces structures :

Code javascript :

```
1 var a = 0, b = 1;
2
3 if (a === 1) {
4     console.log('Je ne serai pas affiché...');
5 } else if (a === 0) {
6     if (b === 1) {
7         console.log('Je serai affiché !');
8     }
9 } else {
10    console.log('Je ne serai pas affiché...');
11 }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

Sélectionner tout

Références

- L'instruction `if...else` (<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Instructions/if...else>).
- Les opérateurs de comparaison (https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Expressions_et_Op%C3%A9rateurs).
- L'algèbre de Boole ([https://fr.wikipedia.org/wiki/Alg%C3%A8bre_de_Boole_\(logique\)](https://fr.wikipedia.org/wiki/Alg%C3%A8bre_de_Boole_(logique))).
- L'ordre de précedence des opérateurs en JavaScript (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Op%C3%A9rateurs/Pr%C3%A9c%C3%A9dence_des_op%C3%A9rateurs).

Exercice

Pour ce premier exercice, il est demandé de déclarer et définir trois variables numériques.

Si elles sont égales, écrire : « Les trois variables sont identiques. »
Si seulement deux sont identiques, écrire : « deux des variables sont de valeurs égales. »
Sinon, écrire : « Les trois variables sont différentes. »

Auteur
Marc Autran

Courage !

Posez toutes vos questions à propos de cet exercice sur cette [discussion du forum](http://www.developpez.net/forums/d1580499/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/facile-indispensable-exercice-1-2-1-structures-conditionnelles-l-instruction-if-else/) (<http://www.developpez.net/forums/d1580499/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/facile-indispensable-exercice-1-2-1-structures-conditionnelles-l-instruction-if-else/>).

[Voir la solution](#)

➤ [Facile][Indispensable] Exercice 1.2.2 : Les structures conditionnelles avec l'instruction "switch"

Mis à jour le 3 mars 2016 par [ABCiWEB](https://www.developpez.com/user/profil/388426/ABCiWEB) (<https://www.developpez.com/user/profil/388426/ABCiWEB>), [autran](https://www.developpez.com/user/profil/863457/autran) (<https://www.developpez.com/user/profil/863457/autran>), [Gnuum](https://www.developpez.com/user/profil/153437/Gnuum) (<https://www.developpez.com/user/profil/153437/Gnuum>)

Objectif

Manipuler les structures conditionnelles de type `switch`.

- Compétences**
- Utilisation des structures conditionnelles de type `if...else`.
 - Utilisation des structures conditionnelles de type `switch`.
 - Compréhension des bases arithmétiques.

Cours

Valeur octale

La base de l'électronique est une suite de tout et de rien : le courant passe ou ne passe pas. Ce qui se traduit en informatique par le fait que le processeur travaille avec une suite de 0 (rien) et de 1 (tout). Ce qui induit que le système décimal (base 10) soit très peu utilisé quand on décrit les fonctionnements internes des ordinateurs. On lui préfère le binaire (base 2) ou l'octal (base 8), ou encore l'hexadécimal (base 16). Pour des raisons historiques de taille d'instruction d'opcode et d'unité d'information, on utilise généralement l'octet pour définir la taille (c.-à-d. le poids) d'un fichier. L'octet est un ensemble de 8 bits, 1 bit étant un 0 ou un 1. Ce qui permet de représenter $2^8=256$ ($2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$) valeurs différentes dans un octet.

Pour pouvoir plus facilement lire les grands nombres d'octets, on a inventé des unités qui elles-mêmes sont fondées sur le système binaire :

- 1 Ko = 2^{10} = 1024 octets (1 Kilooctet)
- 1 Mo = 2^{20} = 1024 x 1024 octets (1 mégaoctet)
- 1 Go = 2^{30} = 1024 x 1024 x 1024 octets (1 gigaoctet)
- 1 To = 2^{40} = 1024 x 1024 x 1024 x 1024 octets (1 téraoctet)

C'est pour cette raison qu'un Kiloctet n'est pas égal à 1000 octets, mais 1024 octets.
À noter que certains considèrent l'appellation Ko pour 1024 octets abusive. Ils préfèrent utiliser Ko pour 1000 octets et Kio (Kibiocet) pour 1024 octets. Cependant, les OS majeurs ne font pas cette distinction.

L'équivalent utilisé pour octet en anglais est *byte*, mais une différence existe entre les deux. En effet, un *byte* ne contient pas la notion d'octal, il a donc également été employé pour des systèmes qui utilisaient des jeux d'instructions différents et où il représentait alors des structures à 7, 9 ou X bits.

Instruction switch

Dans le cas où la condition à tester dans une structure conditionnelle est la valeur d'une seule et même variable, il est plus structurant d'utiliser un `switch` à la place d'un `if...else`. Cela permet d'identifier et comprendre plus rapidement la fonction d'un code, car les possibilités conditionnelles sont restreintes.

Le code :
Code javascript : [Sélectionner tout](#)

```
1 var a = 0;
2
3 if (a === 0) {
4     console.log('a est égal à 0');
5 } else if (a === 1) {
6     console.log('a est égal à 1');
7 } else if (a === 2) {
8     console.log('a est égal à 2');
9 } else if (a === 3) {
10    console.log('a est égal à 3');
11 } else {
12    console.log('a est égal à autre chose');
13 }
```

est équivalent à :
Code javascript : [Sélectionner tout](#)

```
3 switch (a) {
4     case 0:
5         console.log('a est égal à 0');
6         break;
7     case 1:
8         console.log('a est égal à 1');
9         break;
10    case 2:
11        console.log('a est égal à 2');
12        break;
13    case 3:
14        console.log('a est égal à 3');
15        break;
16    default:
17        console.log('a est égal à autre chose');
18 }
```

Le `switch` a d'autres spécificités qui seront vues dans de prochains exercices.

Références

- L'instruction `switch` (<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Instructions/switch>).
- Les bases arithmétiques ([https://fr.wikipedia.org/wiki/Base_\(arithm%C3%A9tique\)](https://fr.wikipedia.org/wiki/Base_(arithm%C3%A9tique))).
- L'octet (<https://fr.wikipedia.org/wiki/Octet>).

Exercice

Le but de cet exercice est de transformer une valeur binaire abrégée en sa valeur non abrégée.
Par exemple : 2K = 2048 (2 x 1024).

Nous allons utiliser trois variables :

- la première `quantity` représente la quantité,
- la deuxième `unit` représente l'unité ("K", "M", "G" ou "T"),
- la troisième `result` contiendra le résultat.

Voici un gabarit de l'exercice :
Code javascript : [Sélectionner tout](#)

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

- 1/ Résoudre l'exercice avec une suite de `if...else`.
- 2/ Résoudre l'exercice avec un `switch` (avec un `break` pour chaque `case`).
- 3/ Faire en sorte que si la variable `unit` est différente des valeurs attendues ("K", "M", "G" ou "T"), la variable `result` soit égale à `false`.

Auteur
Alain Bontemps

Bonne chance !

Posez toutes vos questions à propos de cet exercice sur cette [discussion du forum](http://www.developpez.net/forums/d1580500/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/facile-indispensable-exercice-1-2-2-structures-conditionnelles-l-instruction-switch/) (<http://www.developpez.net/forums/d1580500/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/facile-indispensable-exercice-1-2-2-structures-conditionnelles-l-instruction-switch/>).

[Cacher la solution](#)

Code javascript :

[Sélectionner tout](#)

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

On peut voir qu'il est bien plus facile au premier coup d'œil de déterminer ce qui se passe avec le `switch`.
Le `switch` permet de résoudre ce problème en particulier de manière encore plus élégante, chose que nous verrons dans un prochain exercice.

➤ [\[Intermédiaire\]\[Indispensable\] Exercice 1.2.3 : Les particularités du switch et l'utilisation de la bibliothèque standard](#)
Mis à jour le 3 mars 2016 par [ABCWEB](https://www.developpez.com/user/profil/388426/ABCWEB) (<https://www.developpez.com/user/profil/388426/ABCWEB>), [autran](https://www.developpez.com/user/profil/863457/autran) (<https://www.developpez.com/user/profil/863457/autran>), [Gnuum](https://www.developpez.com/user/profil/153437/Gnuum) (<https://www.developpez.com/user/profil/153437/Gnuum>), [vermine](https://www.developpez.com/user/profil/226061/vermine) (<https://www.developpez.com/user/profil/226061/vermine>).
Objectif
Manipuler les structures conditionnelles de manière avancée.

- Compétences**
- Utilisation des structures conditionnelles.
 - Utilisation avancée de l'instruction `switch`.
 - Utilisation de la bibliothèque standard.
 - Recherche d'informations dans une documentation technique.

Cours
Comportement avancé de l'instruction `switch`
L'instruction `switch` permet un comportement plus fin qu'une simple suite de `if...else` imbriqués testant l'égalité de valeur d'une seule variable, elle permet également une « traversabilité » des conditions ainsi qu'une factorisation du code exécuté en fonction des valeurs grâce à une utilisation judicieuse de l'instruction `break`.

📧 Envoyé par **référence switch**
Si l'on omet une instruction `break`, le script exécutera les instructions pour le cas correspondant et aussi celles pour les cas suivants jusqu'à la fin de l'instruction `switch` ou jusqu'à une instruction `break`.

Note importante : pour différencier rapidement la structure avancée du `switch` (instruction `break` en fonction des besoins), de la structure classique (instruction `break` systématique à la fin de chaque `case`), il est vivement recommandé de remplacer les instructions `break` volontairement omises par le commentaire `//fall through`, excepté lorsque la case est vide, ou encore terminée par un `return` ou un `throw` (que nous verrons dans d'autres occasions), car dans ces cas le code ne prête pas à confusion. Certains outils de programmation javascript comme ESLint renverront une erreur si vous ne respectez pas cette convention. Pour éviter ces contraintes et les risques de confusions, certains considèrent cette structure avancée comme une mauvaise pratique et préconisent de la remplacer par une structure `if/else` correspondante.

Prenons pour exemple le code suivant :

Code JavaScript :

[Sélectionner tout](#)

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
68
```

Les notions de fonction, scope, objet, méthode, prototype et héritage sont des notions importantes en JavaScript qui seront détaillées dans de futurs exercices dédiés. Leur compréhension fine n'est pas indispensable pour le moment.

- Références**
- L'instruction `switch` (<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Instructions/switch>).
 - La bibliothèque standard (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux).
 - La méthode `substr` (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/String/substr).
 - La méthode `toUpperCase` (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/String/toUpperCase).
 - La fonction `parseInt` (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/parseInt).
 - La fonction `isNaN` (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/isNaN).

Exercice

Il s'agit dans cet exercice de comprendre et d'exploiter le comportement de l'instruction `switch` lorsqu'on n'utilise pas l'instruction `break` à la fin de chaque `case`. Parallèlement, nous utiliserons quelques fonctions de la bibliothèque standard du JavaScript pour nous aider. À ce propos, n'hésitez pas à consulter les références proposées ci-dessus pour mieux comprendre le fonctionnement des différentes fonctions et méthodes proposées. Cette consigne sera considérée comme implicite dans les prochains exercices. Un développeur autonome doit être capable de rechercher l'information dont il a besoin par lui même.

1/ Créer une structure de code avec un `switch` qui transforme une variable de type `string` représentant un nombre d'octets abrégé : 150K, 400M, 5G, 3T, en nombre d'octets non abrégé. Le code devra être fonctionnel pour les unités de mesure "K", "M", "G", "T".

Si l'on n'indique pas d'unité, par exemple : `var shortenBytes = '887'`; , on supposera que la variable testée indique des octets.

- Pour rappel :
- 150K = 150 x 1024
 - 400M = 400 x 1024 x 1024
 - 5G = 5 x 1024 x 1024 x 1024
 - 2T = 2 x 1024 x 1024 x 1024 x 1024

Voici un gabarit pour aider à démarrer l'exercice (il est possible le faire évoluer au fur et à mesure de l'énoncé) :

Code JavaScript :

1

2

3

4

5

6

7

8

9

1

0

```
var shortenBytes = '150K';
var letter = // TODO: extract the letter from `shortenBytes` with `substr`.
var bytes;
var number = // TODO: extract a numeric number from `shortenBytes` with `parseInt`.

switch (letter) {
  // TODO: convert `shortenBytes` to `bytes`.
}

console.log(bytes);
```

[Sélectionner tout](#)

2/ Rendre le code insensible à la casse de sorte qu'il pourra fonctionner indifféremment avec des majuscules ou des minuscules, par exemple '250M' ou '250m', '2g' ou '2G', etc. Pour ce faire, on utilisera la méthode `toUpperCase` qui permet de convertir une chaîne de caractères en majuscules.

3/ Sécuriser le script en vérifiant avec `isNaN` que la variable testée respecte le format attendu et peut donc être convertie en valeur numérique. Dans le cas contraire, si par exemple `var shortenBytes = 'two potatoes'`; et plus généralement pour toute valeur non convertible, le code devra afficher le booléen `false`.

Note : durant la phase de développement, ne pas hésiter à utiliser des `console.log` sur les variables pour vérifier que le traitement correspond au résultat attendu. Par exemple :

Code JavaScript :

1

2

3

4

```
// ...
var letter = shortenBytes.substr(/* ... */);
console.log(letter);
// ...
```

[Sélectionner tout](#)

Auteur
Alain Bontemps

Bon boulot !

Posez toutes vos questions à propos de cet exercice sur cette [discussion du forum](http://www.developpez.net/forums/d1580501/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/intermediaire-indispensable-exercice-1-2-3-particularites-switch-l-utilisation-bibliotheq/) (<http://www.developpez.net/forums/d1580501/webmasters-developpement-web/javascript-ajax-typescript-dart/javascript/exercices/intermediaire-indispensable-exercice-1-2-3-particularites-switch-l-utilisation-bibliotheq/>).

[Voir la solution](#)

➤ [Facile][Indispensable] Exercice 1.2.4 : Le typage dynamique

Mis à jour le 3 mars 2016 par [autran](https://www.developpez.com/user/profil/863457/autran/) (<https://www.developpez.com/user/profil/863457/autran/>), [Gnuum](https://www.developpez.com/user/profil/153437/Gnuum) (<https://www.developpez.com/user/profil/153437/Gnuum>).

Objectif

Comprendre le typage dynamique.

- Compétences**
- Utilisation des structures conditionnelles.
 - Récupération du type d'une variable.

Cours

Typage dynamique

Comme nous l'avons vu précédemment, en JavaScript, une variable possède un type. Cependant, contrairement à d'autres langages le type n'est pas imposé par le développeur via des définitions explicites. Ce qui induit que l'interpréteur (système qui traduit le code JavaScript en langage compréhensible par la machine) ne peut vérifier la validité des types et de leur utilisation lors de l'interprétation, mais seulement à l'exécution (quand la variable est définie).

C'est ce qu'on appelle un langage à typage dynamique. Pour ne citer que les principaux avantages et inconvénients, il est plus facile et rapide de travailler avec un typage dynamique, mais le code produit est plus difficile à déboguer et légèrement moins performant.

Récupération du type d'une variable en JavaScript

En JavaScript, il est possible de récupérer le type d'une variable grâce à l'opérateur `typeof` :

Code javascript :

1

2

3

4

5

6

7

8

```
var hi = "Hello !";
console.log(typeof hi); // 'string'
hi = 83110;
console.log(typeof hi); // 'number'
hi = ["H", "e", "l", "l", "o"];
console.log(typeof hi); // 'object'
hi = function() { alert("Hello !"); };
console.log(typeof hi === "function"); // true
```

[Sélectionner tout](#)

Notez qu'en dehors des types primitifs et des fonctions, l'opérateur `typeof` renvoie "object" pour tous les autres types d'objets.

- Références**
- Le typage dynamique (https://fr.wikipedia.org/wiki/Typage_dynamique).
 - L'opérateur `typeof` (https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Op%C3%A9rateurs/L_op%C3%A9rateur_typeof).

Exercice

Déclarer et définir une variable `value` contenant une valeur de type nombre (`number`) ou chaîne de caractères (`string`).

Les valeurs possibles sont `value = 0` ou `value = 1` ou `value = 2` ou `value = '0'` ou `value = '1'` ou `value = '2'`.

- 1/** À l'aide de l'instruction `if...else` afficher "La valeur est une chaîne de caractères" ou "La valeur est un nombre".
- 2/** À l'aide de l'instruction `switch` afficher "La valeur est supérieure à 0" et/ou "La valeur est supérieure à 1" ou "La valeur est égale à 0". Cela doit fonctionner qu'il s'agisse d'un nombre (`number`) ou d'une chaîne de caractères (`string`).

Par exemple pour `var value = 2`; le programme devra afficher :

La valeur est un nombre

La valeur est supérieure à 1

La valeur est supérieure à 0

Auteur
Gnuum

Bonne chance !

