

# JavaScript分享

By SimonAllen

# 目錄(非正式目錄)

- JavaScript簡介
- 本日開發環境
- 型別
- 變數
- 運算子
- 函式
- 物件
- 陣列
- If else
- DOM
- 事件
- 實作

# JavaScript的前世今生

- 1995由Netscape (網景公司，當時瀏覽器廠商之一) 開發
- 為了行銷，故意參考JAVA把名子取名JavaScript
- 目前瀏覽器唯一內建程式語言

# JavaScript的前世今生

- 弱型別，撰寫時不須宣告資料型態
- 由EMCA組織統一版本規格，故JavaScript又稱為ECMAScript
- 2009發布 ECMAScript 5
- 2015發布 ECMAScript 6
- 目前最新版是2016發布的 ECMAScript 7

# 宿主特性

- JavaScript只能運行在使用者的瀏覽器上
- 可以操控HTML和CSS

衍生：使用google V8引擎擴充的JavaScript，打破平台限制的[node.js](https://nodejs.org/)

## 本日開發環境

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
79 $_SESSION['_CAPTCHA']['config'] . serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $separator = ',') {
90 //         $hex_str = preg_replace("/^#/", "", $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return implode($separator, $rgb_array);
105 //     }
106 // }
107
108 // Draw the image
109 // ... (isset($_GET['code']) && !isset($_SESSION['_CAPTCHA']['code'])) {
110 //     $code = $_GET['code'];
111 //     ...
```

# 1. Google Chrome 開發人員工具

瀏覽器右鍵檢查-選擇console

JavaScript程式碼 **console.log(值)**

## 2. Visual Studio Code 編輯器



# 專案資料夾結構

- 新增index.html
- 新增 css 資料夾
- 新增 js 資料夾
- 在 js 資料夾新增app.js

## 檔案總管

### ◀ 已開啟的編輯器

<> index.html

### ◀ LEARNJS

#### ▷ CSS

#### ◀ js

JS app.js

<> index.html

# 引入app.js進html

<> index.html ●

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, init
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10
11   <script src="js/app.js"></script>
12 </body>
13 </html>
```

## 註解

- 不影響程式運作(會佔容量)
- 單行註解：該行說明用//開頭
- 多行註解：/\*開頭與\*/結尾

JS app.js

```
1 // 我是單行註解，不影響程式運作
2
3 /* 我是多行註解，
4  可以打很多字
5  不影響程式運作 */
6
7
```



# 變數

用來儲存資料和進行運算的基本單位

變數會指向記憶體中的程式數據(資料型態)

變數沒有型別，值才有。

變數的預設值是undefined

先宣告，再使用。

# 建立(宣告)變數

宣告關鍵字 變數名稱 ;

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
79 $_SESSION['_CAPTCHA']['confi-1'] . trim(preg_replace('/\\\\\\\\/', '\\', $image_src), '/');
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // ( function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string = 'array') {
90 //         $hex_str = preg_replace('/#/', '', $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) > 0 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //             if ( strlen($hex_str) == 3 ) {
98 //                 $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //                 $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //                 $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //             } else {
102 //                 return false;
103 //             }
104 //         }
105 //         return $return_string == 'implode($separator, $rgb
106 //     }
107 // }
108 // Draw the image
109 // ( function_exists('image') ) {
110 //     $img = imagecreate($width, $height);
111 //     $color = imagecolorallocate($img, $rgb_array['r'], $rgb_array['g'], $rgb_array['b']);
112 //     imagefilledrectangle($img, 0, 0, $width, $height, $color);
113 //     return $img;
114 // }
```

# 建立(宣告)變數

宣告關鍵字

變數名稱

var、let、const

自己取名

# 建立(宣告)變數

var apple;

宣告一個變數 apple



# 變數賦值

已宣告變數 = 值;

# 變數賦值

指定、賦值運算子

已宣告變數

=

值;

宣告過的變數名稱

預給予變數的值

變數賦值

apple = "蘋果";

常用寫法(電腦執行仍拆成宣告、賦值兩個動作)

宣告 變數名稱 = 某個值;

var apple = "蘋果";

# 變數命名學

- 英文數字混用
- 區分大小寫
- 不能數字開頭
- 不能使用-和.
- 不能使用保留字(例如class)
- 盡量不要用特殊符號(例如\$或\_)
- 盡量語意化命名

那我們可以賦予變數哪些值？

```
78 // ... [trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/')];
79 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // ( function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str) {
90 //         $hex_str = strtolower($hex_str);
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // ( function_exists('imagepng') ) {
110 //     $img = imagecreate($width, $height);
111 //     $color = imagecolorallocate($img, $rgb_array['r'], $rgb_array['g'], $rgb_array['b']);
112 //     imagefilledrectangle($img, 0, 0, $width, $height, $color);
113 //     imagepng($img, $image_src);
114 //     imagedestroy($img);
115 // }
```

## 先談談資料型別

```
70 ... . strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
71 ... . trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/');
72
73 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
74
75 return array(
76     'code' => $captcha_config['code'],
77     'image_src' => $image_src
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
```

# JavaScript的資料型別

分成兩大型別

原始型別(Primitive Types)

物件型別(Object Types)



# Object Types物件型別

- array 陣列
- function 函式
- object 物件

與原始型別最大差別在於**物件型別可以自由擴增屬性**

何謂屬性?物件章節會介紹

# Primitive Types原始型別

- undefined 未定義
- null 空
- boolean 布林
- number 數值
- string 字串

# undefined

表示**未定義**

是**所有變數的初始值**，既是值也是型別

剛宣告的變數在我們賦值之前，其值是 undefined，所以開發者最好不要賦值 undefined 給變數。

<> index.html

JS app.js

```
1 // 宣告變數app
2 var app;
3
4 // console.log印出undefined
5 console.log(app);
6
7
8
```

# boolean

表示**true或false**

意即是/否、對/錯、成立/不成立。

```
<> index.html    JS app.js    x
1    var app = true;
2
3    // console.log印出true
4    console.log(app);
5
6    // console.log印出false
7    console.log(10 < 5);
8
```

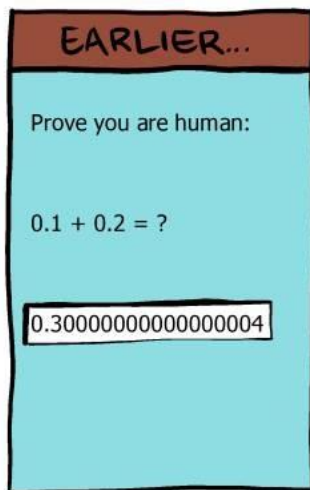
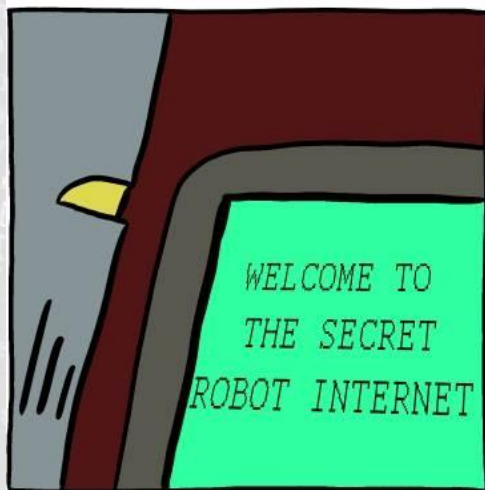
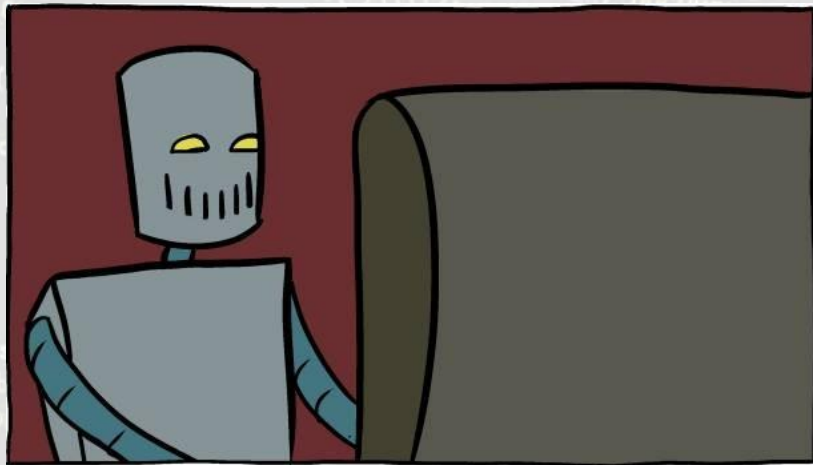
# number

表示**數值**

JS的數字型別只有number，不像其他程式數值有分整數或其他型態

JS的number是浮點數，表示(實際上)有小數點跟在後面。

```
<> index.html JS app.js x
1  var app = 12345;
2
3  // 印出12345
4  console.log(app)
5
6  // 印出0.30000000000000004
7  console.log(0.1 + 0.2);
8
9
```



# null

表示**空值**，既是值也是型別

開發者在宣告變數並要先表示這個變數沒有值時，可以賦值null。

```
<> index.html    JS app.js
1    var app = null;
2
3    // 印出null
4    console.log(app);
5
6
```

null、undefined和0的差異？



- undefined

(此變數) 還沒有給值，所以JavaScript預設其型別與值是undefined

- null

用來表示變數現在空空的沒有值，其型別與值null

- 0

代表型別是number，其值為0，並不是什麼都沒有。



R SATO (佐藤 玲)

@raysato

フォローする



用一張圖理解程式設計新手常搞錯的「0」跟「Null」的分別。



9,648

7,464

リツイート

いいね



4:11 - 2017年2月20日



12



9,648



7,464

# string

表示**字串**

由文字組成，可以用單引號'或雙引號"來包著表示。

```
<> index.html    JS app.js
1  var app = "MAC是修好了沒啊";
2
3  // 印出MAC是修好了沒啊
4  console.log(app)
5
6  // 印出字串123456
7  console.log("123456")
8
9
```

"1000" 和 1000 一樣嗎？

# 運算子

<http://www.runoob.com/js/js-operators.html>

## 常見運算子

名稱	符號	相依順序	範例	結果
加	+	左至右	1+1	2
減	-	左至右	2-1	1
乘	*	左至右	10*2	20
除	/	左至右	15/3	5
餘數	%	左至右	20%7	6

# 常見運算子

名稱	符號	相依順序	範例	結果
賦值	=	右至左	var a = 1;	數值1賦值給a (數值1記憶體位置賦予給a)
存取物件屬性值	.	左至右	var a = {d:100}; console.log(d.a);	存取物件d的屬性a值 100

# 常見運算子

名稱	符號	相依順序	範例	結果
寬鬆比較相等	==	左至右	10 == "10"	回傳true
<b>嚴格</b> 比較相等	===	左至右	10 === "10"	回傳false
寬鬆比較不相等	!=	左至右	5 != "5"	回傳false
<b>嚴格</b> 比較不相等	!==	左至右	5 !== "5"	回傳true

小心JavaScript自動轉換型別的陷阱



常搞混的=、==、===

```
70 // ... [trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/')];
71 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
72
73 return array(
74     'code' => $captcha_config['code'],
75     'image_src' => $image_src
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
```

**=賦值運算子**

JavaScript、jQuery

只要看到=符號，一律理解成

**將=右邊的東西賦予給左邊的東西**

## ==寬鬆比較運算子

將==左邊值與右邊值進行相等比較並回傳布林值(true、false)  
若右邊值與左邊值相似，但型別不同時，JavaScript會將其轉換型別。

## === 嚴格比較運算子

將===左邊值與右邊值**嚴格相等比較**並回傳布林值(true、false)  
若右邊值與左邊值相似，但型別不同時，JavaScript**不會**將其轉換型別。

# 運算子的優先性

- 運算子有分執行優先度
- 運算子優先性高先執行
- 運算子優先性相同，由**相依順序**決定執行方向

小提醒：

=運算子優先性很低，低於+、-、\*、/、%、==、===、!=.....等等

完整運算子優先性表格：<https://goo.gl/T3WfoV>

# 說說這段程式碼的執行順序 1

<> index.html

JS app.js

1 var apple = "我要吃蘋果";

2

3

## 說說這段程式碼的執行順序 2

<> index.html

JS app.js

```
1  var money = 19 + 21;
```

```
2
```

```
3
```

```
4
```

```
5
```

## 說說這段程式碼的執行順序 3

<> index.html

JS app.js

```
1  var dinner = 80;
2  var money = 100 - dinner;
3
4
5
```



## 函式

```
78 // Trim the path to the root of the document
79 $SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string = 'array', $separator = ',') {
90 //         $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // if ( !isset($_GET['code']) ) {
110 //     // ...
111 // }
```

## 函式

- 自己定義的程式包
- 可重複呼叫使用
- 函式內可以放欲執行的程式碼
- 函式內可以宣告變數(區域變數)
- 函式可以接收外部程式傳遞的參數
- 函式可以使用外部程式的變數、函式

# 函式

## 區分成三種

- 函式陳述式(Function Statement)
- 函式表達式(Function Expression)
- 立即執行函式：簡稱IIFE 全名Immediately Invoked Function Expression

# 建立(宣告)函式陳述式

函式關鍵字 函式名稱()

我們的程式碼

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) . '/?_CAPTCHA&t=' .  
79 $_SESSION['_CAPTCHA']['confid']) .  
80 trim(preg_replace('/\\\\\\\\/', '\\', $image_src)) .  
81 return array(  
82     'code' => $captcha_config['code'],  
83     'image_src' => $image_src  
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
```

# 建立(宣告)函式陳述式

```
function sayYes(){  
    console.log("好叻");  
};
```

# 呼叫(執行)函式陳述式

## 函式名稱()

# 呼叫(執行)函式陳述式

# 呼叫(執行)函式陳述式

# 呼叫(執行)函式陳述式

( )

## 括弧



# 建立函式表達式

## 變數 = 函式關鍵字 () { 我們的程式碼

# 建立函式表達式

## 變數 = 函式關鍵字 ( ) { 我們的程式碼

# 建立函式表達式

## 變數 = 函式關鍵字 ( ) { 我們的程式碼

# 建立函式表達式

```
var sayHi = function () {  
  console.log("晚上好");  
};
```

# 呼叫(執行)函式表達式

## 變數名稱();

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
79 $_SESSION['_CAPTCHA']['confi-13'] . trim(preg_replace('/\\|\\|/', '/', $image_src));
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // ( function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_array = true, $separator = ',') {
90 //         $hex_str = preg_replace("/^#-s+ra-f)/", "", $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $return_array ? implode($separator, $rgb_array) : $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // ( function_exists('imagepng') ) {
110 //     $img = imagecreatefrompng($image_src);
111 //     imagepng($img, $image_src);
112 // }
```

# 呼叫(執行)函式表達式

sayHi();

# 呼叫(執行)函式表達式

sayHi

()

變數名稱 括弧

```
$_SESSION['_CAPTCHA']['confi-13
79 $length = strlen( realpath($_SERVER['DOCUMENT_ROOT']) );
80 $captcha = trim(preg_replace('/\\\\\\\\/', '\\', $image_src));
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return
90 //     {
91 //         $hex_str = preg_replace("/[^0-9a-fA-F]/", '', $hex_str); // Gets a proper hex string
92 //         $rgb_array = array();
93 //         if ( strlen($hex_str) == 6 ) {
94 //             $color_val = hexdec($hex_str);
95 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
96 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
97 //             $rgb_array['b'] = 0xFF & $color_val;
98 //         } elseif ( strlen($hex_str) == 3 ) {
99 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
100 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
101 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
102 //         } else {
103 //             return false;
104 //         }
105 //         return $return_string;
106 //     }
107 // }
108 // Draw the image
109 // if ( !empty($_GET['code']) ) {
110 //     $code = $_GET['code'];
111 // }
```

## 函式陳述式與表達式的差異

	函式陳述式	函式表達式
函式命名	需有名子	匿名函式
呼叫函式	透過函式的名子	透過變數的名子
函式本身可以當成值賦值給變數	否	可
函式本身可以當成值傳給其他程式	否	可

## 函式內的變數-區域變數

1. 在函式內宣告的變數，即是區域變數。
2. 只要是函式，都可以在函式內宣告變數。
3. 區域變數只存在該函式內，不同函式內的變數彼此獨立。
4. 區域變數可與函式外變數取相同名子，彼此互不影響。
5. 區域變數一定要使用**宣告關鍵字**宣告，否則會變全域變數。

# 宣告區域變數

<> index.html

JS app.js

```
1  
2  
3 var yes = "Yessss!"  
4
```

```
5 function sayYes(){  
6     var yes = "No"  
7     console.log(yes);  
8 } ;  
9
```

```
10  
11 sayYes()  
12  
13
```

全域

區域



# 宣告區域變數

<> index.html

JS app.js

```
1
2
3   var yes = "Yesssss!"
4
5   function sayYes(){
6       var yes = "No"
7   }
8   ;
9
10  sayYes()
11  console.log(yes);    結果是?
12
13
```

# 宣告區域變數

<> index.html

JS app.js

```
1  
2  
3 var yes = "Yesssss!"
```

```
4  
5 function sayYes(){  
6   console.log(yes);  
7 } ;
```

```
8  
9 sayYes() 結果是?  
10
```

# 外部查找

當函式內找不到指定的區域變數  
函式會向函式外的程式查找

```
78 // ... [trim(preg_replace('/\\|\\|\\|/', '/', $image_src), '/'));
79 $_SESSION['_CAPTCHA']['config']
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85
86 }
87
88 // ...
89
90 function_exists('hex2rgb') ) {
91     function hex2rgb($hex_str, $return_string = true) {
92         $hex_str = preg_replace('/\\|\\|\\|/', '/', $hex_str);
93         $rgb_array = array();
94         if ( strlen($hex_str) == 6 ) {
95             $color_val = hexdec($hex_str);
96             $rgb_array['r'] = ($color_val >> 0x10);
97             $rgb_array['g'] = ($color_val >> 0x08);
98             $rgb_array['b'] = ($color_val >> 0x00);
99         } elseif ( strlen($hex_str) == 3 ) {
100             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
101             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
102             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
103         } else {
104             return false;
105         }
106         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
107     }
108 }
109
110 // Draw the image
111 if ( !isset($_GET['code']) ) {
112     // ...
113 }
```

# 宣告區域變數

<> index.html

JS app.js

```
1  
2  
3 var yes = "Yesssss!"  
4
```

```
5 function sayYes(){  
6     console.log(yes);  
7 } ;  
8
```

```
9 sayYes() 結果是?  
10
```

# 傳遞參數

1. 只要是函式都可以從函式外傳遞參數進函式內
2. 參數命名與變數命名規範相同
3. 參數只存在函式內且不影響函式外變數
4. 傳遞參數與呼叫函式一起執行
5. 呼叫函式為傳遞參數，參數在函式內為undefined

函式陳述式

函式關鍵字 函式名稱(參數){  
運用這個參數做一些事  
};

函式表達式

變數 = 函式關鍵字 (參數){  
運用這個參數做一些事  
};

## 以函式陳述式為例-傳遞參數

```
function sayYes(x){  
    console.log(x);  
};
```

```
sayYes("Yesssss!");
```

```
var yes = "Yesssss!"
```

```
function sayYes(x){  
    console.log(x);  
};
```

```
sayYes(yes);
```

## 陣列

```
70 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
71 $_SESSION['_CAPTCHA']['config'] . serialize($captcha_config);
72
73 return array(
74     'code' => $captcha_config['code'],
75     'image_src' => $image_src
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
```



- 櫃子
- 需用「,」符號分隔內容
- 雜湊陣列-每層都可以存放不同型別資料

## 建立陣列

- 陣列實體語法
- []
- 取用第一個[0]

## 建立陣列

- 陣列實體語法
- []
- 取用第一個[0]

## 陣列練習

```
[ 2018,  
  Monospace,  
  function(){  
    console.log(free)  
  },  
]
```

## 物件

```
70 $SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
71
72 return array(
73     'code' => $captcha_config['code'],
74     'image_src' => $image_src
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
```

## 名子(屬性)與值的組合

## 流程判斷

```
70 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
71 // ... trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/');
72
73 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
74
75 return array(
76     'code' => $captcha_config['code'],
77     'image_src' => $image_src
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
```

如果成立就...

If(true){

//執行內容

}



如果成立就...否則就..

If(true){

//執行內容

} else {

//執行內容

}

如果成立就...否則如果..否則就..

If(true){

//執行內容

} else if{

//執行內容

} else {

//執行內容

}

# DOM

```
78 // ... trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/');
79 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string) {
90 //         $hex_str = preg_replace("/[A0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if (strlen($hex_str) == 6) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif (strlen($hex_str) == 3) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // ... (isset($_GET['code']) && !isset($_SESSION['_CAPTCHA']['code'])) {
110 //     $code = $_GET['code'];
111 //     ...
112 }
```

# 何謂DOM物件?

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .  
79 $_SESSION['_CAPTCHA']['config'];  
80  
81 return array(  
82     'code' => $captcha_config['code'],  
83     'image_src' => $image_src  
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
```

# 理解DOM樹狀圖(1)

```
$_SESSION['_CAPTCHA']['confi-12'] = trim(preg_replace('/\\\\\\\\/', '\\\\', $image_src)) . '?_CAPTCHA&amp;t=' .  
return array(  
    'code' => $captcha_config['code'],  
    'image_src' => $image_src  
);  
  
function_exists('hex2rgb') ) {  
    function hex2rgb($hex_str, $return_string = false, $separator = ',') {  
        $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string  
        $rgb_array = array();  
        if (strlen($hex_str) == 6) {  
            $color_val = hexdec($hex_str);  
            $rgb_array['r'] = 0xFF & ($color_val >> 0x10);  
            $rgb_array['g'] = 0xFF & ($color_val >> 0x08);  
            $rgb_array['b'] = 0xFF & $color_val;  
        } elseif (strlen($hex_str) == 3) {  
            $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));  
            $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));  
            $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));  
        } else {  
            return false;  
        }  
        return $return_string ? implode($separator, $rgb_array) : $rgb_array;  
    }  
}
```

## 理解DOM樹狀圖(2)

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
79 $_SESSION['_CAPTCHA']['confi-13'] . trim(preg_replace('/\\\\\\\\/', '\\', $image_src), '/') );
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // ( function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string = false, $separator = ',') {
90 //         $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // ( function_exists('imagepng') ) {
110 //     $img = imagepng($image_src);
111 // }
```

## 理解DOM樹狀圖(3)

```
$_SESSION['_CAPTCHA']['confi-13
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&
79 $SESSION['_CAPTCHA']['confi-13
80 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&
81 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&
82 return array(
83     'code' => $captcha_config['code'],
84     'image_src' => $image_src
85 );
86 }
87
88 // ... function_exists('hex2rgb') ) {
89 // ... hex2rgb($hex_str, $return_string = false, $separator = ',') {
90 // ... preg_replace("/[A0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 // ... array();
92 // ... strlen($hex_str) == 6 ) {
93 // ... $color_val = hexdec($hex_str);
94 // ... $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 // ... $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 // ... $rgb_array['b'] = 0xFF & $color_val;
97 // ... strlen($hex_str) == 3 ) {
98 // ... $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 // ... $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 // ... $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 // ... return false;
102 // ... return $return_string ? implode($separator, $rgb
103 // ... Draw the image
104 // ... ( $_GET['
105 // ...
```

## 依照ID去獲取節點



## 依照class去獲取節點

```
78 $SESSION['_CAPTCHA']['config'] = array( 'code' => $captcha_config['code'],  
79 'image_src' => $image_src );  
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
```

## 事件

```
70 $SESSION['_CAPTCHA']['config'] = serialize($captcha_config);
71
72 return array(
73     'code' => $captcha_config['code'],
74     'image_src' => $image_src
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
```

# 什麼是事件?

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .  
79 $_SESSION['_CAPTCHA']['config'];  
80  
81 return array(  
82     'code' => $captcha_config['code'],  
83     'image_src' => $image_src  
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
```

# 事件監聽

## 練習時間

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '?_CAPTCHA&t=' .
79 $_SESSION['_CAPTCHA']['config'] . serialize($captcha_config);
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 // ( function_exists('hex2rgb') ) {
89 //     function hex2rgb($hex_str, $return_string = true, $separator = ',') {
90 //         $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
91 //         $rgb_array = array();
92 //         if ( strlen($hex_str) == 6 ) {
93 //             $color_val = hexdec($hex_str);
94 //             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95 //             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96 //             $rgb_array['b'] = 0xFF & $color_val;
97 //         } elseif ( strlen($hex_str) == 3 ) {
98 //             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99 //             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100 //             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101 //         } else {
102 //             return false;
103 //         }
104 //         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105 //     }
106 // }
107
108 // Draw the image
109 // ( function_exists('imagecreate') ) {
110 //     $img = imagecreate(100, 100);
111 //     $color = imagecolorallocate($img, $rgb_array['r'], $rgb_array['g'], $rgb_array['b']);
112 //     imagefill($img, 0, 0, $color);
113 //     imagepng($img, $image_src);
114 //     imagedestroy($img);
115 // }
```





# 關於我



## 參考資源

```
$_SESSION['_CAPTCHA']['config']
return array(
    'code' => $captcha_config['code'],
    'image_src' => $image_src
);

function_exists('hex2rgb') ) {
    function hex2rgb($hex_str, $return_string = false, $separator = ',') {
        $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper hex string
        $rgb_array = array();
        if (strlen($hex_str) == 6) {
            $color_val = hexdec($hex_str);
            $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
            $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
            $rgb_array['b'] = 0xFF & $color_val;
        } elseif (strlen($hex_str) == 3) {
            $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
            $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
            $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
        } else {
            return false;
        }
        return $return_string ? implode($separator, $rgb_array) : $rgb_array;
    }

    // Draw the image
    if (isset($_GET['code'])) {
        $code = $_GET['code'];
        $image_src = hex2rgb($code, true, '.');
        $image_src = $image_src . '?_CAPTCHA&t=' . time();
    }
}
```