

## 22/01/13 - Manquejades del PHP in\_array

Sempre dic que PHP és un llenguatge que m'agrada molt, i que em diverteixo molt treballant-hi i que un projecte Internet es pot fer diverses vegades més ràpid que en Java.

Ara bé, de la mateixa manera també dic que és ple de problemes, problemes de disseny i errors estúpids.

Avui explicaré quelcom prou poc intuïtiu i que us pot portar a greus problemes al vostre codi.

L'ús d' in\_array

Observeu aquest codi:

```
<?php
Array$st_array = array(
    'nom1' => 0, Array
    'nom2' => 1, Array
    'nom3' =>
    '2', Array
    'nom4' => 'Catalunya
Lliure');
ArrayArrayecho 'Cas 0:'.in_array('0',
$st_array)."\n";
Arrayecho 'Cas 1:'.in_array('1',
$st_array)."\n";
Arrayecho 'Cas 2:'.in_array('2',
$st_array)."\n";
Arrayecho 'Cas n:'.in_array('n',
$st_array)."\n";
Arrayecho 'Cas z:'.in_array('z',
$st_array)."\n";
Arrayecho 'Cas
Catalunya:'.in_array('Catalunya', $st_array)."\n";
Arrayecho
'Cas num no:'.in_array(3, $st_array)."\n";
Arrayecho 'Cas num
sí:'.in_array(1, $st_array)."\n";
ArrayArray$st_array2 =
array(2, 'a', 'Catalunya Lliure');
ArrayArrayecho 'Cas
1a:'.in_array(3, $st_array2)."\n";
Arrayecho 'cas
2a:'.in_array(2, $st_array2)."\n";
Arrayecho 'cas
3a:'.in_array('2', $st_array2)."\n";
Arrayecho 'Cas
4a:'.in_array('Catalunya', $st_array2)."\n";
Arrayecho 'Cas
5a:'.in_array('Catalunya Lliure', $st_array2)."\n";
```

Qualsevol desenvolupador de PHP diria que el resultat que espera és diferent del que rebrà.

Veiem el resultat:

```

carles@ninja: ~/Escriptori/codi
Fitxer Edita Visualitza Cerca Terminal Ajuda
carles@ninja:~/Escriptori/codi$ php -f test_array_keys_loose.php
Cas 0:1
Cas 1:1
Cas 2:1
Cas n:1
Cas z:1
Cas Catalunya:1
Cas num no:
Cas num sí:1
Cas 1a:
cas 2a:1
cas 3a:1
Cas 4a:
Cas 5a:1
carles@ninja:~/Escriptori/codi$

```

Els resultats que he obtingut són els mateixos en sistemes Debian amb 5.3.3:

```

PHP 5.3.3-7+squeeze14 with Suhosin-Patch (cli) (built: Aug 6 2012 14:18:06)
Copyright (c) 1997-2009 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2010 Zend Technologies
with Xdebug v2.1.0, Copyright (c) 2002-2010, by Derick Rethans

```

En Ubuntu amb 5.3.10:

```

PHP 5.3.10-1ubuntu3.4 with Suhosin-Patch (cli) (built: Sep 12 2012 18:59:41)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2012 Zend Technologies
with Xdebug v2.2.1, Copyright (c) 2002-2012, by Derick Rethans

```

I en PHP 5.4.10 i en 5.5.0.a.2 executats en aquesta sandbox:

<http://sandbox.onlinephpfunctions.com/>

El resultat és l'esperat\* en els casos en que el valor de l'array és numèric, però sembla totalment buggy en el cas de string (el que coneixem com Map a Java o hash a Python).

Veient els exemples hom pensaria que el problema es produeix si els valors assignats a barregen valors numèrics i string, però també si l'array té claus alfanumèriques (?nom1? => ?0'). Aquest últim cas es veu clarament amb l'exemple de cercar ?Catalunya' al primer i al segon cas.

Però si tingués un bon amic com el meu amic Edu, us faria adonar que no es tracta de si la clau és alfanumèrica o automàtica, si no que el fet que hi hagi un 0 (zero) com a valor integer, és el que provoca aquest comportament esbojarrat i imprevisible del PHP.

Mirant la documentació del mètode loose de comparació de PHP observem on és el problema.

Encara que sigui contra intuïtiu, en el mètode loose qualsevol cadena de text és == a 0 (zero integer).

\* Nota: Dic que és l'esperat donat que la documentació de PHP per a in\_array ja

indica que "Searches	haystack
for	needle
using loose comparison unless	strict
is set.", és a dir, que fa servir el mètode de comparació loose, en que es considera igual 1 i "1" per exemple.	

I aquest és un dels grans problemes de PHP, que degut al seu comportament intern, a les conversions entre tipus que fa automàticament, viola els principis de programació de fiabilitat\*\*, i porta a situacions en que és altament probable que els programadors generin bugs o es generin bugs que només passen en situacions molt específiques.

\*\* Quan dic que viola el principis de programació de fiabilitat de resultats, em refereixo a coses com:

Expressió	Resultat
"Catalunya" == 0	TRUE
"0" == 0	TRUE
"Catalunya" == "0"	FALSE

O sigui "Catalunya" == 0 i 0 == "0" però "Catalunya" és diferent de "0". Epic fail.

Haurem d'utilitzar === que fa la comprovació de tipus també.

La solució universal que farà que els dos casos d'in\_array funcionin correctament és emprar el mètode estricte.

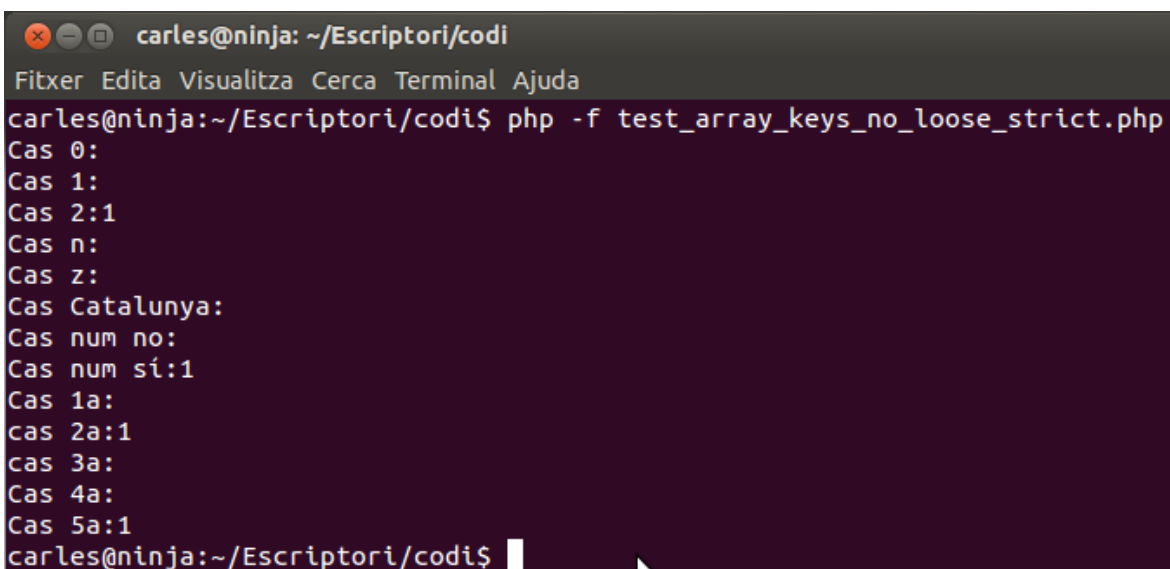
Codi que funciona segons s'espera:

```

<?php Arraydefine('METODE_STRICTE', true); ArrayArray$st_array
= array( 'nom1' => 0, Array 'nom2' => 1, Array 'nom3' =>
'2', Array 'nom4' => 'Catalunya Lliure');ArrayArrayecho 'Cas
0:'.in_array('0', $st_array, METODE_STRICTE)."\n";Arrayecho
'Cas 1:'.in_array('1', $st_array,
METODE_STRICTE)."\n";Arrayecho 'Cas 2:'.in_array('2',
$st_array, METODE_STRICTE)."\n";Arrayecho 'Cas
n:'.in_array('n', $st_array, METODE_STRICTE)."\n";Arrayecho
'Cas z:'.in_array('z', $st_array,
METODE_STRICTE)."\n";Arrayecho 'Cas
Catalunya:'.in_array('Catalunya', $st_array,
METODE_STRICTE)."\n";Arrayecho 'Cas num no:'.in_array(3,
$st_array, METODE_STRICTE)."\n";Arrayecho 'Cas num
sí:'.in_array(1, $st_array,
METODE_STRICTE)."\n";ArrayArray$st_array2 =
array(2,'a','Catalunya Lliure');ArrayArrayecho 'Cas
1a:'.in_array(3, $st_array2, METODE_STRICTE)."\n";Arrayecho
'cas 2a:'.in_array(2, $st_array2,
METODE_STRICTE)."\n";Arrayecho 'cas 3a:'.in_array('2',
$st_array2, METODE_STRICTE)."\n";Arrayecho 'Cas
4a:'.in_array('Catalunya', $st_array2,
METODE_STRICTE)."\n";Arrayecho 'Cas 5a:'.in_array('Catalunya
Lliure', $st_array2, METODE_STRICTE)."\n";

```

I podeu observar el resultat:



```

carles@ninja: ~/Escriptori/codi
Fitxer Edita Visualitza Cerca Terminal Ajuda
carles@ninja:~/Escriptori/codi$ php -f test_array_keys_no_loose_strict.php
Cas 0:
Cas 1:
Cas 2:1
Cas n:
Cas z:
Cas Catalunya:
Cas num no:
Cas num sí:1
Cas 1a:
cas 2a:1
cas 3a:
Cas 4a:
Cas 5a:1
carles@ninja:~/Escriptori/codi$

```

Pareu atenció a que amb el mètode estricte si cerqueu ?1' (string) i a l'array hi ha 1 (integer) in\_array us retornarà FALSE.

Molt important a tenir en compte que tot el que rebeu de \$\_GET o \$\_POST sigui el que sigui us arribarà com un string, o un array d'strings, malgrat en el formulari hagin inserit valors numèrics com 2014.

Nota per a usuaris novells: Jo he utilitzat una constant amb el valor booleà TRUE per a fer servir el mètode estricte però prodrieu fer servir directament true.

Traduir a l'Anglès. Translate to English	Compartir:
--	------------