

Movian - Bug #2531

parseInt parses "08" and "09" incorrectly

01/29/2015 12:34 AM - Fredrik Lundmark

Status:	Invalid	Start date:	01/29/2015
Priority:	Normal	Due date:	
Assignee:	Andreas Smas	% Done:	0%
Category:	General	Estimated time:	0.00 hour
Target version:		Platform:	Linux
Found in version:	4.9.120.g5a131		

Description

parseInt parses the strings "08" and "09" incorrectly to the value 0. I tried numbers 0-110 in both JS and ecmaScript. Don't know if it acts funny if you get to higher numbers than that.

Code to reproduce in ecmaScript:

```
var i, numbers;
```

```
numbers = ["00", "01", "02", "03", "04", "05", "06", "07", "08", "09", "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"];  
for(i = 0; i < numbers.length; i++) {  
  console.log("\n loop number: " + i + "\nstring: " + numbers[i] + "\nparseInt string: " + parseInt(numbers[i]));  
}
```

Code to reproduce in javascript:

```
var i, numbers;  
numbers = ["00", "01", "02", "03", "04", "05", "06", "07", "08", "09", "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"];  
for(i = 0; i < numbers.length; i++) {  
  showtime.trace("\n loop number: " + i + "\nstring: " + numbers[i] + "\nparseInt string: " + parseInt(numbers[i]));  
}
```

The interesting bit is:

```
test      [DEBUG]: loop number: 8  
string: 08  
parseInt string: 0  
test      [DEBUG]: loop number: 9  
string: 09  
parseInt string: 0
```

I ran a quick check on numbers 0 - 110 and counted parseInt string:

```
hp@G74Sx ~/tmp/DreamFilm $ grep -i "parseInt string: 0" parseInt_JS | wc -l  
3  
hp@G74Sx ~/tmp/DreamFilm $ grep -i "parseInt string: 0" parseInt_ecma | wc -l  
3
```

"00" is parsed as 0 above, which is correct. Deduct that expected result from the results and we have 2 numbers that parse incorrectly as 0; "08", and "09".

Note: This bug is present on the PS3 as well.

History

#1 - 01/29/2015 01:08 AM - Fredrik Lundmark

Ok, adding radix solved this issue. Apparently radix is not optional. You can close this issue.

#2 - 01/29/2015 07:48 AM - Andreas Smas

- Status changed from New to Invalid

Yes, having a 0 as prefix it parses is as an octal value (so 0 ... 7 is valid but 8 and 9 are not)

Files

parseInt_ecma	16.9 KB	01/28/2015	Fredrik Lundmark
parseInt_JS	13.3 KB	01/28/2015	Fredrik Lundmark
showtime-1.log	23.7 KB	01/28/2015	Fredrik Lundmark
showtime-0.log	23.9 KB	01/28/2015	Fredrik Lundmark