**6 kyu**

**Maximum Product of Parts**

7096% of 2639 of63[marbiru](https://www.codewars.com/users/marbiru)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/maximum-product-of-parts/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/maximum-product-of-parts/solutions/python)

[Discourse (15)](https://www.codewars.com/kata/maximum-product-of-parts/discuss/python)

* Add to Collection
* |
* Share this kata:

The written representation of a number (with 4 or more digits) can be split into three parts in various different ways. For example, the written number 1234 can be split as [1 | 2 | 34] or [1 | 23 | 4] or [12 | 3 | 4].

Given a written number, find the highest possible product from splitting it into three parts and multiplying those parts together. For example:

* product of [1 | 2 | 34] = 1 \* 2 \* 34 = 68
* product of [1 | 23 | 4] = 1 \* 23 \* 4 = 92
* product of [12 | 3 | 4] = 12 \* 3 \* 4 = 144

So maximumProductOfParts(1234) = 144

For a longer string there will be many possible different ways to split it into parts. For example, 8675309 could be split as:

* [8 | 6 | 75309]
* [867 | 530 | 9]
* [8 | 67 | 5309]
* [86 | 75 | 309]

or any other option that splits the string into three parts each containing at least one digit.

<https://www.codewars.com/kata/maximum-product-of-parts/python>

1. **def** maximum\_product\_of\_parts(n):
2. max\_prod = 0
3. s = str(n)
5. **for** i **in** range(1, len(s)-1):
6. **for** j **in** range(i+1, len(s)):
7. a = s[0:i]
8. b = s[i:j]
9. c = s[j:]
11. max\_prod = max(max\_prod, int(a) \*int(b)\* int(c))
13. **return** max\_prod

16. **print**(maximum\_product\_of\_parts(123456))