

# Table of Contents

<code>/home/runner/work/PA_TEST_ACTION/PA_TEST_ACTION/test.py</code> .....	2
<code>/home/runner/work/PA_TEST_ACTION/PA_TEST_ACTION/main.py</code> .....	3

## **/home/runner/work/PA\_TEST\_ACTION/PA\_TEST\_ACTION/test.py**

```
def test():
```

```
    return "a"
```

-> Returns a Unicode string containing the given Unicode string .

```
def delete_double_spaces(text):
```

```
    return text.replace(' ', '')
```

-> Replaces double spaces in a string with the given text .

```
def delete_doublon_array(array):
```

```
    return list(set(array))
```

-> Returns a list of zubute vectors

```
def delete_doublon_string(string):
```

```
    return ''.join(set(string))
```

-> Returns a copy of the string with all the second ellipsis .

```
def find_value_in_array(array, value):
```

```
    return array.index(value)
```

-> Find the array value in array for a given value

```
def find_value_in_string(string, value):
```

```
    return string.index(value)
```

-> Returns the index of the value of the given string

## Documentation

```
def get_promotion_name(promotion_name):
```

```
    return promotion_name.split('_')[0]
```

-> Translates a snake\_case promotion name to the name in question

```
def apply_pourcent_in_array(array, pourcent):
```

```
    return [x * pourcent for x in array]
```

```
# Modifier la police
```

-> Traverses the given array in a pretty list while assumes the Colour is being used .

```
def bonjour_test():
```

```
    print("bonjour")
```

-> Prints a Bonjourn that evaluates the given string to a terminal the use

**/home/runner/work/PA\_TEST\_ACTION/PA\_TEST\_ACTION/main.py**

```
def max_number(a, b):  
  
    return max(a, b)  
  
if __name__ == '__main__':  
  
    print( (lambda a,b: a*b )(5,4) - True)
```

-> The maximum number of bytes given two strings .