Function Examples

What Would Python Display?

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<table>
<thead>
<tr>
<th>This expression</th>
<th>Evaluates to</th>
<th>Interactive Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>print(5)</td>
<td>None</td>
<td>5</td>
</tr>
<tr>
<td>print(print(5))</td>
<td>None</td>
<td>5</td>
</tr>
<tr>
<td>delay(delay)(6)</td>
<td>6</td>
<td>delayed 6</td>
</tr>
<tr>
<td>print(delay(print)(4))</td>
<td>None</td>
<td>delayed 4</td>
</tr>
</tbody>
</table>
### What Would Python Print?

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

<table>
<thead>
<tr>
<th>This expression</th>
<th>Evaluates to</th>
<th>Interactive Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add(pirate(3)(square)(4), 1)</code></td>
<td>17</td>
<td><code>Matey 17</code></td>
</tr>
<tr>
<td><code>pirate(pirate(pirate))(5)(7)</code></td>
<td>Error</td>
<td><code>Error</code></td>
</tr>
</tbody>
</table>

A function that always returns the identity function

<table>
<thead>
<tr>
<th>A function that always returns the identity function</th>
</tr>
</thead>
</table>
| `import add, mul
def square(x):
    return mul(x, x)` |
| `def pirate(arggg):
    print('matey')` |
| `def plunder(arggg):
    return arggg
    return plunder` |

### Implementing a Function

Implementing a Function

<table>
<thead>
<tr>
<th>Implementing a Function</th>
</tr>
</thead>
</table>
| `def remove(n, digit):
    """Return all digits of non-negative N that are not DIGIT, for some non-negative DIGIT less than 10."""
    kept, digits = 0, 0
    while n > 0:
        last = n // 10, n % 10
        if last != digit:
            kept = kept + last**10
        digits = digits + 1
    return kept + last**10**digits

>>> remove(231, 3)
 21
>>> remove(243132, 2)
 21 231

Read the description
Verify the examples & pick a simple one
Read the template
Implement without the template, then change your implementation to match the template.
OR
If the template is helpful, use it.
Annotate names with values from your chosen example
Write code to compute the result
Did you really return the right thing?
Check your solution with the other examples
Implementing a Function

```python
def remove(n, digit):
    """Return all digits of non-negative N
    that are not DIGIT, for some non-negative DIGIT less than 10.
    >>> remove(231, 3)
    21
    >>> remove(243132, 2)
    23132
    ""
    kept, digits = 0, 0
    while n > 0:
        n, last = n // 10, n % 10
        if last != digit:
            kept = kept * 10 + last
        digits += 1
    return round(kept * 10 ** (digits-1))
```

Function Decorators

(Demo)

```python
from functools import wraps

@trace1
def triple(x):
    return 3 * x

# is identical to

def triple(x):
    return 3 * x
triple = trace1(triple)
```

Decorators