

Ruby Hash - Addition, Deletion, Selection

In this challenge, we will show you ways in which we can add key-value pairs to Hash objects, delete keys from them, and retain them based on a logic.

Consider the following **Hash** object:

```
h = Hash.new
h.default = 0
```

- A new key-value pair can be added using or the **store** method

```
h[key] = value
```

or

```
h.store(key, value)
```

- An existing key can be deleted using the **delete** method

```
h.delete(key)
```

- For destructive selection and deletion, we can use **keep_if** and **delete_if** as seen in [Array-Selection](#)

```
> h = {1 => 1, 2 => 4, 3 => 9, 4 => 16, 5 => 25}
=> {1 => 1, 2 => 4, 3 => 9, 4 => 16, 5 => 25}
> h.keep_if {|key, value| key % 2 == 0} # or h.delete_if {|key, value| key % 2 != 0}
=> {2 => 4, 4 => 16}
```

Note

For non-destructive selection or rejection, we can use **select**, **reject**, and **drop_while** similar to [Array-Selection](#)

In this challenge, a hash object called **hackerrank** is already created. You have to add

- A key-value pair [543121, 100] to the **hackerrank** object using **store**
- Retain all key-value pairs where keys are Integers (clue : is_a? Integer)
- Delete all key-value pairs where keys are even-valued.