

```
[num for num in fibs if is_even(num)]
```

```
[1, 1, 2, 3, 5, 8, 13, 21, 34]
```



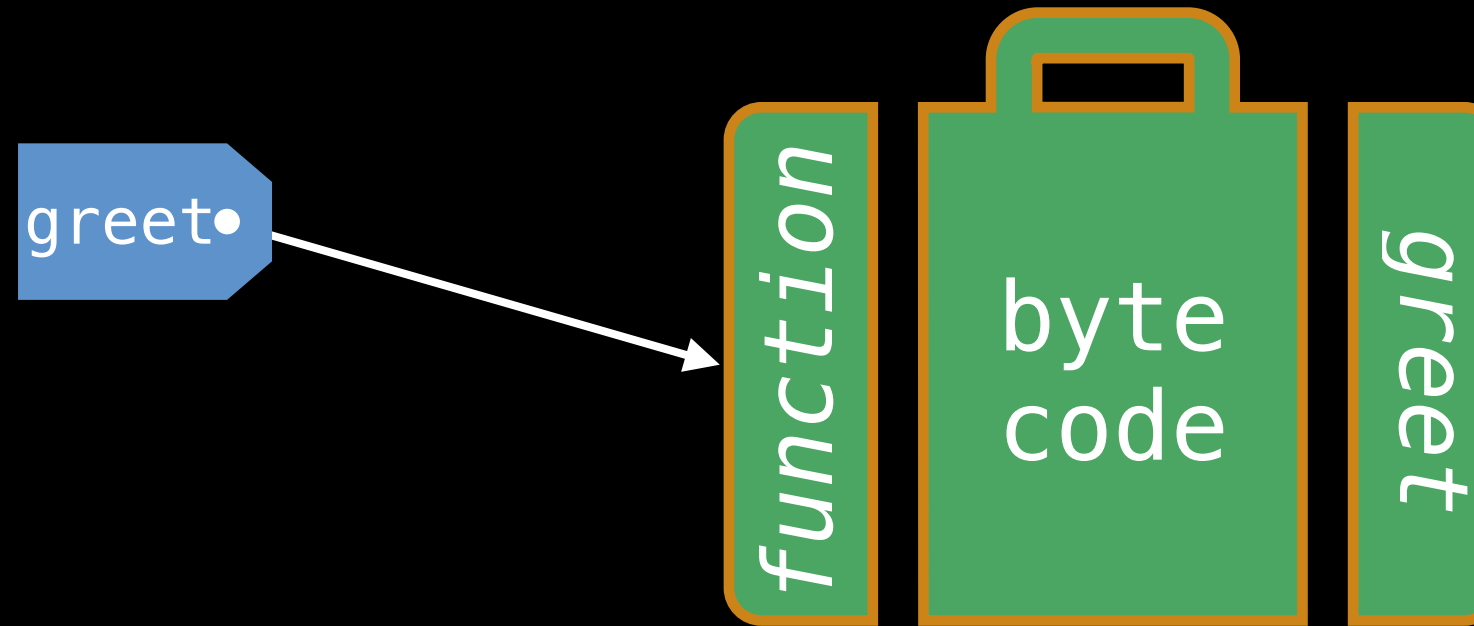
The diagram illustrates the process of filtering even numbers from a list of Fibonacci numbers. At the top, a list of Fibonacci numbers is shown: [1, 1, 2, 3, 5, 8, 13, 21, 34]. Below this list, a yellow oval acts as a central hub. Red lines connect the odd numbers (1, 1, 3, 5, 13, 21) to this oval, while green lines connect the even numbers (2, 8, 34) to it. Below the oval is a blue rectangular box containing the text 'filter(is_even, fibs)'. From the bottom center of this box, a green line leads to a green dot. From this dot, three green arrows point downwards to the filtered result: '< 2, 8, 34 >'. The entire diagram is set against a black background.

```
filter(is_even, fibs)
```

```
< 2, 8, 34 >
```

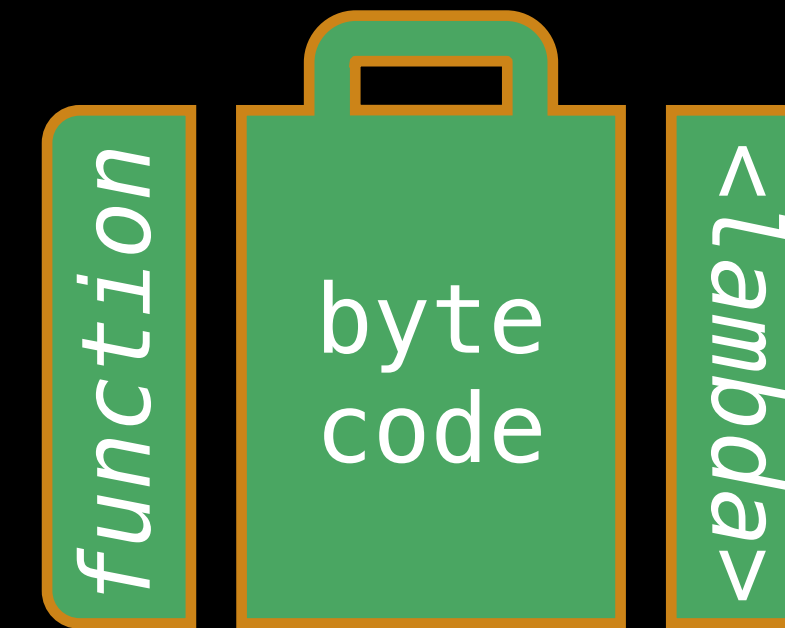
Defined Functions vs. Lambdas

```
def greet():  
    print("Hi!")
```



`def` binds a name to a function object

```
lambda val: val ** 2  
lambda x, y: x * y  
lambda pair: pair[0] * pair[1]
```



`lambda` only creates a function object

```
(lambda x: x > 3)(4) # => True
```

Creates a function object and immediately call it