

# Python Filter()

The `filter()` function **selects elements from an iterable** (list, tuple etc.) based on the output of a function.

The function is applied to each element of the iterable and if it returns `True`, the element is selected by the `filter()` function.

## Example

```
# returns True if the argument passed is even

def check_even(number):
    if number % 2 == 0:
        return True

    return False

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# if an element passed to check_even() returns True, select it
even_numbers_iterator = filter(check_even, numbers)

# converting to list
even_numbers = list(even_numbers_iterator)

print(even_numbers)

# Output: [2, 4, 6, 8, 10]
```

## filter() Syntax

The syntax of `filter()` is:

```
filter(function, iterable)
```

## filter() Arguments

The `filter()` function takes two arguments:

- **function** - a function
- **iterable** - an iterable like [sets](#), [lists](#), [tuples](#) etc.

## filter() Return Value

The `filter()` function returns an iterator.

**Note:** You can easily convert iterators to sequences like lists, tuples, strings etc.

# Example 1: Working of filter()

```
letters = ['a', 'b', 'd', 'e', 'i', 'j', 'o']

# a function that returns True if letter is vowel
def filter_vowels(letter):
    vowels = ['a', 'e', 'i', 'o', 'u']
    return True if letter in vowels else False

filtered_vowels = filter(filter_vowels, letters)

# converting to tuple
vowels = tuple(filtered_vowels)
print(vowels)
```

Here, the `filter()` function extracts only the vowel letters from the `letters` list. Here's how this code works:

- Each element of the `letters` list is passed to the `filter_vowels()` function.
- If `filter_vowels()` returns `True`, that element is extracted otherwise it's filtered out.

**Note:** It's also possible to filter lists using a loop, however, using the `filter()` function is much more cleaner.

## Example 2: Using Lambda Function Inside filter()

```
numbers = [1, 2, 3, 4, 5, 6, 7]

# the lambda function returns True for even numbers
even_numbers_iterator = filter(lambda x: (x%2 == 0), numbers)

# converting to list
even_numbers = list(even_numbers_iterator)

print(even_numbers)
```

## Example 3: Using None as a Function Inside filter()

```
# random list
random_list = [1, 'a', 0, False, True, 'o']

filtered_iterator = filter(None, random_list)

#converting to list
filtered_list = list(filtered_iterator)

print(filtered_list)
```