

all() and any()

`all()` and `any()` are built-in functions in Python that allow us to conveniently check for boolean matching in an iterable. `all()` will return `True` if all elements in an iterable are `True`. It is the same as this function code:

```
def all(iterable):  
    for element in iterable:  
        if not element:  
            return False  
    return True
```

`any()` will return `True` if any of the elements in the iterable are `True`. It is equivalent to the following function code:

```
def any(iterable):  
    for element in iterable:  
        if element:  
            return True  
    return False
```

Let's see a few examples of these functions. They should be fairly straightforward:

```
In [4]: lst = [True, True, False, True]
```

```
In [5]: all(lst)
```

```
Out[5]: False
```

Returns `False` because not all elements are `True`.

```
In [6]: any(lst)
```

```
Out[6]: True
```

Returns `True` because at least one of the elements in the list is `True`

There you have it, you should have an understanding of how to use `any()` and `all()` in your code.