

05-Useful-Operators

January 7, 2021

1 Useful Operators

There are a few built-in functions and “operators” in Python that don’t fit well into any category, so we will go over them in this lecture, let’s begin!

1.1 range

The range function allows you to quickly *generate* a list of integers, this comes in handy a lot, so take note of how to use it! There are 3 parameters you can pass, a start, a stop, and a step size. Let’s see some examples:

```
[1]: range(0,11)
```

```
[1]: range(0, 11)
```

Note that this is a **generator** function, so to actually get a list out of it, we need to cast it to a list with **list()**. What is a generator? Its a special type of function that will generate information and not need to save it to memory. We haven’t talked about functions or generators yet, so just keep this in your notes for now, we will discuss this in much more detail in later on in your training!

```
[3]: # Notice how 11 is not included, up to but not including 11, just like slice_
    ↪ notation!
    list(range(0,11))
```

```
[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
[4]: list(range(0,12))
```

```
[4]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
[6]: # Third parameter is step size!
    # step size just means how big of a jump/leap/step you
    # take from the starting number to get to the next number.

    list(range(0,11,2))
```

```
[6]: [0, 2, 4, 6, 8, 10]
```

```
[7]: list(range(0,101,10))
```

```
[7]: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

1.2 in operator

We've already seen the **in** keyword during the for loop, but we can also use it to quickly check if an object is in a list

```
[21]: 'x' in ['x','y','z']
```

```
[21]: True
```

```
[22]: 'x' in [1,2,3]
```

```
[22]: False
```

1.3 not in

We can combine **in** with a **not** operator, to check if some object or variable is not present in a list.

```
[1]: 'x' not in ['x','y','z']
```

```
[1]: False
```

```
[2]: 'x' not in [1,2,3]
```

```
[2]: True
```

1.4 min and max

Quickly check the minimum or maximum of a list with these functions.

```
[26]: mylist = [10,20,30,40,100]
```

```
[27]: min(mylist)
```

```
[27]: 10
```

```
[44]: max(mylist)
```

```
[44]: 100
```

1.5 random

Python comes with a built in random library. There are a lot of functions included in this random library, so we will only show you two useful functions for now.

```
[29]: from random import shuffle
```

```
[35]: # This shuffles the list "in-place" meaning it won't return  
# anything, instead it will effect the list passed
```

```
shuffle(mylist)
```

```
[36]: mylist
```

```
[36]: [40, 10, 100, 30, 20]
```

```
[39]: from random import randint
```

```
[41]: # Return random integer in range [a, b], including both end points.  
randint(0,100)
```

```
[41]: 25
```

```
[42]: # Return random integer in range [a, b], including both end points.  
randint(0,100)
```

```
[42]: 91
```

1.6 input

```
[43]: input('Enter Something into this box: ')
```

```
Enter Something into this box: great job!
```

```
[43]: 'great job!'
```