Assignment Operators

This lesson showcases Python's various assignment operators and their purpose.

We'll cover the following

This is a category of operators which is used to assign values to a variable. The `=` operator we've used until now is an assignment operator, but not the only one.

Here's a list of all the assignment operators supported in Python:

Operator	Purpose	Notation
`=`	Assign	In-fix
`+=`	Add and Assign	In-fix
`-=`	Subtract and Assign	In-fix
`*=`	Multiply and Assign	In-fix
`/=`	Divide and Assign	In-fix
`//=`	Divide, Floor, and Assign	In-fix
`**=`	Raise power and Assign	In-fix
`%=`	Take Modulo and Assign	In-fix

`|=`, `&=`, `^=`	OR/AND/XOR and Assign	In-fix
`>>=`	Right-shift and Assign	In-fix
`<<=`	Left-shift and Assign	In-fix

Assigning Values

Let's go through a few examples to see how values are assigned to variables.

Variables are **mutable**, so we can change their values whenever we want!

```
year = 2019
print(year)

year = 2020
print(year)

year = year + 1  # Using the existing value to create a new one
print(year)
```

One thing to note is that when a variable, `first`, is assigned to another variable, `second`, its value is **copied** into `second`. Hence, if we later change the value of `first`, `second` will remain unaffected:

```
first = 20
second = first
first = 35  # Updating 'first'

print(first, second)  # 'second' remains unchanged
```

The Other Operators

Below, we can see some of the assignment operators we talked about in action:

```
num = 10
print(num)

num += 5
```

```
print(num)

num -= 5
print(num)

num *= 2
print(num)

num /= 2
print(num)

num **= 2
print(num)

# Try all the others here!
```

In the next lesson, we'll study **logical operators**.