

Immutability

Cannot mutate or cannot change

Lets create a variable 'name' and set it to Sam

```
In [1]: name = "Sam"
```

If we wanted to change Sam to Pam, we could do the following:

```
In [2]: name[0]= 'P'
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20960\10177451.py in <module>
----> 1 name[0]= 'P'

TypeError: 'str' object does not support item assignment
```

We cannot do what we attempted above as strings are immutable However, we can merge two strings together like so:

```
In [4]: name[1:]
```

```
Out[4]: 'am'
```

```
In [5]: last_letters = name[1:]
```

```
In [6]: last_letters
```

```
Out[6]: 'am'
```

```
In [7]: 'P' + last_letters
```

```
Out[7]: 'Pam'
```

```
In [11]: x = 'Hello World'
```

```
In [25]: x = x + " it is beautiful outside"
```

```
In [26]: x
```

```
Out[26]: 'Hello World \n it is beautiful outside it is beautiful outside it is beautiful outside'
```

We can also multiple string to get mutiple, like so, we assign z to letter then multiple it by 10

```
In [24]: letter = 'z'
```

```
In [21]: letter * 10
```

```
Out[21]: 'zzzzzzzzzz'
```

So pictured below, we can add 2+3 to get 5 However, we can set 2 as a string with '2' and 3 as a string with '3', instead of adding them together, we put the strings together to get 23

```
In [27]: 2 + 3
```

```
Out[27]: 5
```

```
In [28]: '2' + '3'
```

```
Out[28]: '23'
```

We can also modify the string either by x.(tab) or we could do 'x.upper()'

```
In [29]: x = 'Hello World'
```

```
In [31]: x.upper()
```

```
Out[31]: 'HELLO WORLD'
```

However, this does not set the 'Hello World' as capital, we would have to set it as capital with the following:

```
In [32]: x = x.upper()
```

```
In [33]: x
```

```
Out[33]: 'HELLO WORLD'
```

```
In [34]: x = 'Hello World'
```

```
In [35]: x
```

```
Out[35]: 'Hello World'
```

There is also a lowercase method like so:

```
In [36]: x = x.lower()
```

```
In [37]: x
```

```
Out[37]: 'hello world'
```

There is also a split method which allows us to create a list off of a string like so:

```
In [38]: x.split()
```

Out[38]:

```
In [39]: x = 'Hi this is a string'
```

```
In [40]: x.split()
```

Out[40]: ['Hi', 'this', 'is', 'a', 'string']

Everything above was split by the white space by default

However, we could also split it by a letter, like so:

```
In [42]: x.split('i')
```

Out[42]: ['H', ' th', 's ', 's a str', 'ng']

Strings - FAQ

Are strings mutable? Strings are not mutable (meaning you cant use indexing to change invidivual elements of a string)

How do I create comments in my code? You can use the hashtag to create comments in your code