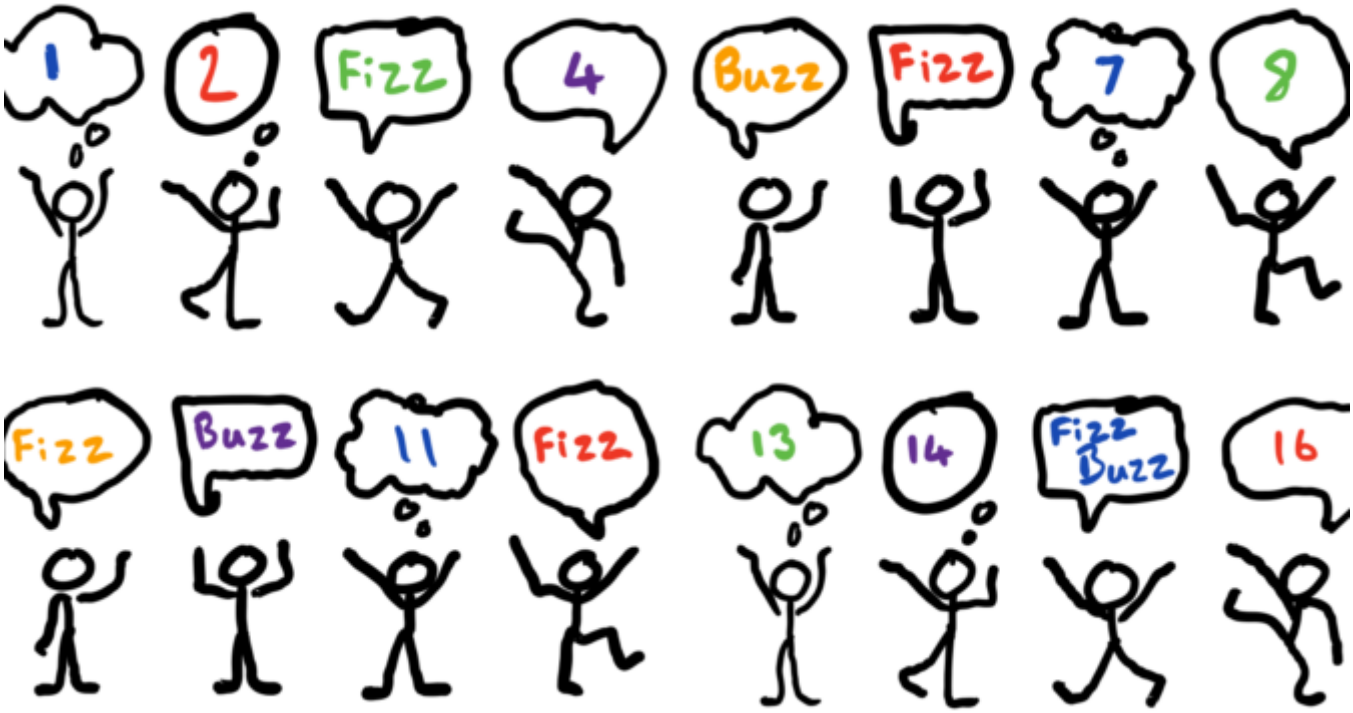
Fizz Buzz Algorithm.



For a series of a number from 1 to 10.

1 2 3 4 5 6 7 8 9 10

Fizz and Buzz refer to any number that's a multiple of 3 and 5 respectively. In other words, if a number is divisible by 3, it is substituted with fizz; if a number is divisible by 5, it is substituted with buzz. If a number is simultaneously a multiple of 3 AND 5, the number is replaced with "fizz buzz." In essence, it emulates the famous children game "fizz buzz".

To work on this problem, open up Google colab to create a new .ipynb file and initialize an array like below:

```
## for example
let_number = [1,2,3,4,5]
## here 3 is fizz and 5 is buzz
```

To find all the fizz and buzz, we must iterate through the array and check which numbers are fizz and which are buzz. To do this, create a for loop to iterate through the array we have initialised:

```
for num in let_number {
// Body and calculation goes here
```

```
}
```

After this, we can simply use the "if else" condition and module operator ie - % to locate the fizz and buzz.

```
for num in let_number:
 if num % 3 == 0:
  print(num, "fizz")
 else:
  print(num)
```

Great! You can go to the debug console in google colab to see the output. You will find that the "fizzes" have been sorted out in your array.

For the Buzz part, we will use the same technique. Let's give it a try before scrolling through the code — you can check your results against this code once you've finished doing this.

```
for num in let_number:
  if num % 3 == 0:
    print(num, "fizz")
  elif num % 5 == 0:
    print(num, "buzz")
  else:
    print(num)
```

Check the output!

It's rather straight forward — you divided the number by 3, fizz and divided the number by 5, buzz. Now, increase the numbers in the array

```
let_number = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
```

We increased the range of numbers from 1-10 to 1-15 in order to demonstrate the concept of a "fizz buzz." Since 15 is a multiple of both 3 and 5, the number should be replaced with "fizz buzz."

Try for yourself and check the answer! Here is the solution:

```
for num in let_number:
 if num % 3 == 0 and num % 5 == 0:
    print(num, "fizz buzz")
 elif num % 3 == 0:
    print(num, "fizz")
 elif num % 5 == 0:
```

```
    print(num, "buzz")
 else:
    print(num)
```

```
Wait...it's not over though! The whole pur
customize the runtime correctly. Imagine i
1-100. The compiler will check each number
divisible by 3 or 5. It would then run thr
the numbers are divisible by 3 and 5. The
through each number in the array twice — i
3 first and then run it by 5. To speed up
code to divide the numbers by 15 directly.

Here is the final code:
```

Wait...it's not over though! The whole purpose of the algorithm is to customize the runtime correctly. Imagine if the range increases from 1-15 to 1-100. The compiler will check each number to determine whether it is divisible by 3 or 5. It would then run through the numbers again to check if the numbers are divisible by 3 and 5. The code would essentially have to run through each number in the array twice — it would have to runs the numbers by 3 first and then run it by 5. To speed up the process, we can simply tell our code to divide the numbers by 15 directly.

Here is the final code:

```
let_number = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

for num in let_number:
 if num % 15 == 0:
    print(num, "fizz buzz")
 elif num % 3 == 0:
    print(num, "fizz")
 elif num % 5 == 0:
    print(num, "buzz")
 else:
    print(num)
```

```
1
2
3 fizz
4
5 buzz
6 fizz
7
8
9 fizz
10 buzz
11
```

```
12 fizz
13
14
15 fizz buzz
```

✓ 0s completed at 11:10 AM