

Replace method

★ Every rule should have a unique find string.

Can't have: $[[\text{'F'}, \text{'FF'}], [\text{'F'}, \text{'+F'}]]$
need to be different

★ L System string: 'F+F'

Rules: $[[\text{'F'}, \text{'FF'}]]$

Start 'F+F'
↳ 'FF+F'

how I move on to next character (+).

$\Rightarrow$ if I don't move on:

$F + F$
 $FF + F$
 $FFF + F$
 $FFFF + F$
 $\vdots$

Don't want to
do this

Only apply one rule per character in string,
then I move on.

Cher
↓
Curr string = 'F+F+F'
Self.rules = $\left[\underbrace{[-', 'F+F']}_{\text{rule}}, \underbrace{['F', 'FF']}_{\text{rule}} \right]$

```
def replace(self, currString): 'F++F++F'  
    newString = ''
```

```
    for char in currString:
```

```
        # check to see if char matches
```

```
        # a find string in rules
```

```
        found = False
```

```
        for rule in self.rules:
```

```
            # loop thru rules
```

```
            if char == rule[0]:
```

```
                # do we have a match?
```

```
                newString += rule[1]
```

```
                found = True
```

```
                break
```

```
        if not found:
```

```
            newString += char
```

```
    return newString
```

find str

replace str