

```
[ ]: 1. procedural style- syntax, type, logic- direct approach
      2. functional style- inline
      3. object oriented - class-> object
```

```
L=[]
def fx(a):
    return a*10
for var in range(10):
    r=fx(var)
    L.append(r)
print(L)
```

```
[ ]: list(map(lambda a:a+10, range(10))) # functional
```

```
[ ]: class cname:
      def ...
```