```
# @Title:   ICPC-Cheat-Sheet
# @Author:  Regan O'Donnell
# @DoFB:    2/23/2024

# @Desc: A "python 101" intended manuscript containing all basic to complex things (that I could think of) you can do in python
# This sheet also acts as a refresher if I ever take long breaks from python
# This file is not intended to be "ran", as most of it is psuedo code
# Use the "playground.py" file to copy and paste code and mess with it

# ~ Code Format (CTRL+F)~
# 1.) Literals
# 2.) if, else-if, else, and, or
# 3.) Loops
# 4.) class

# Literals
255, 0b11111111, 0o377, 0xff # Integers (decimal, binary, octal, hex)
123.0, 1.23                  # Float
7 + 5j, 7j                   # Complex
'a', '\141', '\x61'          # Character (literal, octal, hex)
'\n', '\\', '\'', '\"'       # Newline, backslash, single quote, double quote
"string\n"                   # String of characters ending with newline
"hello"+"world"              # Concatenated strings
True, False                  # bool constants, 1 == True, 0 == False
[1, 2, 3, 4, 5]              # List
['meh', 'foo', 5]            # List
(2, 4, 6, 8)                 # Tuple, immutable
{'name': 'a', 'age': 90}     # Dict
{'a', 'e', 'i', 'o', 'u'}    # Set
None                         # Null var

# if, else-if, else
if True and True:
    print("if")
elif False or False:
    print("else-if")
else:
    print("else")

# Loops
```

```
# go through all elements (while)
i = 0
while i < len(str):
    i += 1

# go through all elements (for)
for i in range (len(str)):
    print(i)

# go through all elements (for + list)
for i in array:
    print(i) # will print element at array index

# reverse the order of an array: [1,2,3] -> [3,2,1]
nums = [1,2,3,4,5,9,6,7,8,5]

for i in nums: # prints the list
    print(i, end="")

l, r = 0, len(nums) - 1
while l < r: # reverses elements
  nums[l], nums[r] = nums[r], nums[l]
  l += 1
  r -= 1

for i in nums: # prints the list
    print(i, end="")

# class
class MyClass:
    def __init__(self, public_var, private_var):
        self.public_var = public_var
        self.__private_var = private_var

    def public_method(self):
        print("This is a public method.")
        self.__private_method()

    def __private_method(self):
        print("This is a private method.")
```

```
    def print_vars(self):
        print("Public variable:", self.public_var)
        print("Private variable:", self.__private_var)

    def __del__(self):
        print("Destructor called, object deleted.")

# Example usage:
obj = MyClass("Public", "Private")
obj.public_method()
obj.print_vars()
```