Python 3 Cheatsheet: Core Syntax

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
```

Loops

Go through all elements

```
# Equivalent loops
i = 0
while i < len(str):
  i += 1

for i in range(len(message)):
  print(i)
```

Common Patterns

```
# Range: range(start, stop, step)
for a in range(0,3):        # 0, 1, 2
for a in reversed(range(0,3)): # 2, 1, 0
for i in range(3,-1,-1):    # 3, 2, 1, 0

# Tilde (~) indexing
for i in range(len(A)//2):  # A = [0,1,2,3,4,5]
  print(A[i])   # 0, 1, 2
  print(A[~i])  # 5, 4, 3 (Simultaneous front/back access)
```

Strings

```
# Search & Check
'pen' in 'pencil'       # True (Membership)
s.find('x')             # Index or -1
s.rfind('x')            # Last index or -1
s.startswith("sub")     # True/False
s.endswith("sub")       # True/False
s.isalnum()             # Alpha-numeric
s.isalpha()             # Alphabetical
s.isdigit()             # Digit

# Modification
s.strip()               # Remove whitespace (lstrip/rstrip)
s.replace('old', 'new') # Replace substring
s.upper()               # Uppercase
s.lower()               # Lowercase
s.swapcase()            # Invert case

# Splitting & Joining
"a b c".split()         # ['a', 'b', 'c'] (Default: whitespace)
"a,b,c".split(',')      # ['a', 'b', 'c']
" ".join(['a','b'])     # "a b"

# Formatting
ord('A')                # 65
chr(65)                 # 'A'
"meh" * 2               # "mehmeh"
f"Hi {name}"            # f-string
"Val: {}".format(x)     # format() method

# Common Patterns
s = s[::-1]             # Reverse string (via slicing)
is_pal = s == s[::-1]   # Palindrome check
from collections import Counter
Counter(s1) == Counter(s2) # Anagram check
```

Slicing

`sliceable[start:stop:step]`

```
p = ['P','y','t','h','o','n']
p[0]              # 'P'
p[0:5]            # ['P','y','t','h','o']
p[0:5:2]          # ['P', 't', 'o']
p[5:0:-1]         # ['n', 'o', 'h', 't', 'y']
p[::-1]           # Reverse list/string

# Slice Assignment
p[2:4] = ['t','r'] # Replace index 2,3
```