Variables

```
a = 1      # integer
b = 1.1    # float
c = 1 + 2j # complex number (a + bi)
d = “a”    # string
e = True   # boolean (True / False)
```

Strings

```
x = “Python”
len(x)
x[0]
x[-1]
x[0:3]

# Formatted strings
name = f”{first} {last}”

# Escape sequences
\” \’ \\ \n

# String methods
x.upper()
x.lower()
x.title()
x.strip()
x.find(“p”)
```

```
x.replace(“a”, “b”)
“a” in x
```

Type Conversion

```
int(x)
float(x)
bool(x)
string(x)
```

Falsy Values

```
0
“”
[]
```

Conditional Statements

```
if x == 1:
    print(“a”)
elif x == 2:
    print(“b”)
else:
    print(“c”)

# Ternary operator
x = “a” if n > 1 else “b”

```

```
# Chaining comparison operators
if 18 <= age < 65:
```

Loops

```
for n in range(1, 10):
    print(n)

while n < 10:
    print(n)
    n += 1
```

Functions

```
def increment(number, by=1):
    return number + by

# Keyword arguments
increment(2, by=1)

# Variable number of arguments
def multiply(*numbers):
    for number in numbers:
        print(number)


multiply(1, 2, 3, 4)

# Variable number of keyword arguments
```

```
def save_user(**user):
    ...


save_user(id=1, name="Mosh")
```

Lists

```
# Creating lists
letters = ["a", "b", "c"]
matrix = [[0, 1], [1, 2]]
zeros = [0] * 5
combined = zeros + letters
numbers = list(range(20))

# Accessing items
letters = ["a", "b", "c", "d"]
letters[0]  # "a"
letters[-1] # "d"

# Slicing lists
letters[0:3]   # "a", "b", "c"
letters[:3]    # "a", "b", "c"
letters[0:]    # "a", "b", "c", "d"
letters[:]     # "a", "b", "c", "d"
letters[::2]   # "a", "c"
letters[::-1]  # "d", "c", "b", "a"

# Unpacking
first, second, *other = letters

# Looping over lists
for letter in letters:
    ...

for index, letter in enumerate(letters):
    ...

```

```
# Adding items
letters.append("e")
letters.insert(0, "-")

# Removing items
letters.pop()
letters.pop(0)
letters.remove("b")
del letters[0:3]

# Finding items
if "f" in letters:
    letters.index("f")

# Sorting lists
letters.sort()
letters.sort(reverse=True)

# Custom sorting
items = [
    ("Product1", 10),
    ("Product2", 9),
    ("Product3", 11)
]

items.sort(key=lambda item: item[1])

# Map and filter
prices = list(map(lambda item: item[1], items))

```

```
expensive_items = list(filter(lambda item: item[1] >= 10, items))

# List comprehensions
prices = [item[1] for item in items]
expensive_items = [item for item in items if item[1] >= 10]

# Zip function
list1 = [1, 2, 3]
list2 = [10, 20, 30]
combined = list(zip(list1, list2)) # [(1, 10), (2, 20)]
```

Tuples

```
point = (1, 2, 3)
point(0:2)     # (1, 2)
x, y, z = point
if 10 in point:
    ...

# Swapping variables
x = 10
y = 11
x, y = y, x
```

Arrays

```
from array import array

numbers = array("i", [1, 2, 3])
```

Sets

```
first = {1, 2, 3, 4}
second = {1, 5}

first | second  # {1, 2, 3, 4, 5}
first & second  # {1}
first - second  # {2, 3, 4}
first ^ second  # {2, 3, 4, 5}

if 1 in first:
    ...
```

Dictionaries

```
point = {"x": 1, "y": 2}
point = dict(x=1, y=2)
point["z"] = 3
if "a" in point:
    ...
point.get("a", 0)   # 0
del point["x"]
for key, value in point.items():
   ...

# Dictionary comprehensions
values = {x: x * 2 for x in range(5)}
```

Generator Expressions

```
values = (x * 2 for x in range(10000))
len(values)  # Error
for x in values:
```

Unpacking Operator

```
first = [1, 2, 3]
second = [4, 5, 6]
combined = [*first, "a", *second]

first = {"x": 1}
second = {"y": 2}
combined = {**first, **second}
```

Source: https://programmingwithmosh.com/python/python-3-cheat-sheet/