

Python Fundamentals CHEATSHEET

Quick Reference -- Variables - Operators - Control Flow - Strings - OOP

DATA TYPES

<code>int</code>	Whole numbers: 42, -7, 0
<code>float</code>	Decimals: 3.14, -0.5
<code>str</code>	Text: "Hello", 'World'
<code>bool</code>	True or False
<code>None</code>	No value / empty box

VARIABLES

Assignment: `variable_name = value`

```
player = "Asad" # str
score = 1500 # int
ratio = 0.75 # float
alive = True # bool
guild = None # None
```

Watch Out: = stores. == compares. Never confuse them!

ARITHMETIC

<code>+ - *</code>	Add / Sub / Multiply
<code>/</code>	Division -- always float
<code>//</code>	Floor div -- int (10//3=3)
<code>%</code>	Remainder (10%3=1)
<code>**</code>	Power (2**8=256)

SHORTCUTS

```
x += 5 # x = x + 5
x -= 3 # x = x - 3
x *= 2 # x = x * 2
x //= 2 # x = x // 2
x **= 2 # x = x ** 2
```

COMPARISON

<code>== !=</code>	Equal / Not equal
<code>> <</code>	Greater / Less than
<code>>= <=</code>	Gte / Lte

LOGICAL

<code>and</code>	Both must be True
<code>or</code>	At least one True
<code>not</code>	Flips True to False

```
can_enter = has_ticket and is_adult
            and not is_banned
```

COMMENTS

```
# Single-line comment
x = 10 # inline
"""
Multi-line
"""
```

Note: Explain WHY, not WHAT.

IF / ELIF / ELSE

```
if marks >= 90:
    grade = 'A'
elif marks >= 75:
    grade = 'B'
elif marks >= 60:
    grade = 'C'
else:
    grade = 'F'
```

Note: Colon after condition. Indent 4 spaces.

FOR LOOP

```
for i in range(5): # 0-4
    print(i)
for i in range(1, 6): # 1 to 5
    print(i)
# With index:
for i, v in enumerate(lst):
    print(i, v)
```

WHILE LOOP

```
count = 0
while count < 5:
    print(count)
    count += 1
# break -- exit loop
# continue -- skip to next
```

FUNCTIONS

```
def greet(name):
    return f"Hi {name}!"

result = greet("Asad")
```

```
# Default parameter:
def greet(name, msg="Hi"):
    print(f"{msg} {name}")
```

Note: return sends value back. Parameters are local.

LISTS

```
scores = [85, 92, 78]
scores[0] # 85 (first)
scores[-1] # 78 (last)
scores[0:2] # [85, 92]
scores.append(61) # add end
scores.remove(92) # del value
len(scores) max(scores) sum(scores)
```

Python Fundamentals CHEATSHEET

Page 2 of 2 -- Dictionaries - Strings - Input - OOP

DICTIONARIES

```
p = {"name": "Asad", "lvl": 12}
p["name"] # access
p["coins"] = 500 # add/update
del p["lvl"] # remove
p.get("xp", 0) # safe get
for k,v in p.items():
    print(k,":",v)
```

STRINGS & F-STRINGS

```
msg = " Hello World "
msg.upper() # UPPERCASE
msg.strip() # remove spaces
msg.replace("o","0") # replace
"a,b,c".split(",") # ['a','b','c']
len("hello") # 5

f"Hi {name}, score:{score:.2f}"
```

USER INPUT

```
name = input("Your name: ")
age = int(input("Age: "))
prc = float(input("Price: "))
```

Watch Out: input() always returns string -- convert with int() or float()!

COMMON ERRORS

SyntaxError	missing colon after if/for/def
IndentError	bad indentation
TypeError	str + int directly
NameError	undefined variable
IndexError	list index out of bounds

OOP -- CLASSES & OBJECTS

```
class Student: # blueprint
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def introduce(self):
        print(f"Hi I'm {self.name}")
```

```
s1 = Student("Asad", 15) # object
s1.introduce()
```

Note: class = blueprint. object = real thing made from blueprint.

OOP -- INHERITANCE

```
class Animal:
    def __init__(self, name):
        self.name = name
    def breathe(self):
        print(f"{self.name}: breathing")
```

```
class Dog(Animal): # inherits Animal
    def bark(self):
        print(f"{self.name}: WOOF!")
```

```
dog = Dog("Bruno")
dog.breathe() # inherited!
dog.bark() # Dog's own
```

OOP -- super() & __str__

```
class Dog(Animal):
    def __init__(self, name, breed):
        super().__init__(name) # parent first
        self.breed = breed

    def __str__(self):
        return f"Dog({self.name}, {self.breed})"
```

```
d = Dog("Bruno", "Lab")
print(d) # Dog(Bruno, Lab)
```

TYPE CONVERSION

int(x)	str/float to integer
float(x)	str/int to decimal
str(x)	number to string
bool(x)	0/None/" = False

range() Quick Guide	Useful Built-ins	4 OOP Pillars
range(5) -> 0 1 2 3 4 range(1,6) -> 1 to 5 range(0,10,2) -> 0 2 4 6 8	print() len() type() input() min() max() sum() abs() round() sorted() enumerate() isinstance(x, int)	Encapsulation -- bundle data+methods Inheritance -- child gets parent code Polymorphism -- same name diff result Abstraction -- hide complexity