

Python Cheat Sheet

Variables and Data Types

Variables (no declaration needed)

```
name = "Alice"
```

```
age = 25
```

```
height = 5.6
```

```
is_student = True
```

Data types

```
str_var = "Hello"    # String
```

```
int_var = 42         # Integer
```

```
float_var = 3.14     # Float
```

```
bool_var = True      # Boolean
```

```
list_var = [1, 2, 3] # List
```

```
tuple_var = (1, 2, 3) # Tuple
```

```
dict_var = {"key": "value"} # Dictionary
```

```
set_var = {1, 2, 3}  # Set
```

Print and Input

```
print("Hello World")
```

```
print(f"Name: {name}, Age: {age}") # f-strings
```

```
name = input("Enter your name: ")
```

If Statements

```
if age >= 18:
```

```
    print("Adult")
```

```
elif age >= 13:
```

```
    print("Teenager")
```

```
else:
```

```
    print("Child")
```

Loops

For loop

```
for i in range(5):  
    print(i)
```

```
for item in [1, 2, 3]:  
    print(item)
```

```
# While loop  
count = 0  
while count < 5:  
    print(count)  
    count += 1
```

Lists

```
# Creating and modifying  
lst = [1, 2, 3, 4, 5]  
lst.append(6)      # Add to end  
lst.insert(0, 0)   # Insert at index  
lst.remove(3)      # Remove first occurrence  
lst.pop()          # Remove last item  
lst[0] = 10        # Change item  
len(lst)           # Length
```

Dictionaries

```
# Creating and accessing  
person = {"name": "Alice", "age": 25}  
person["name"]     # Access value  
person["city"] = "NYC" # Add new key-value  
del person["age"]   # Delete key  
person.keys()      # Get all keys  
person.values()     # Get all values  
person.get("name", "Unknown") # Safe access
```

Sets

```
# Creating and operations  
s1 = {1, 2, 3, 4}  
s2 = {3, 4, 5, 6}  
s1.add(5)          # Add element  
s1.remove(1)       # Remove element
```

```
s1 & s2          # Intersection
s1 | s2          # Union
s1 - s2          # Difference
```

Functions

```
def greet(name, greeting="Hello"):
    return f'{greeting}, {name}!'
```

```
def add_numbers(*args):
    return sum(args)
```

```
def person_info(**kwargs):
    for key, value in kwargs.items():
        print(f'{key}: {value}')
```

```
# Lambda functions
square = lambda x: x**2
```

List Comprehensions

```
# Basic list comprehension
squares = [x**2 for x in range(10)]
```

```
# With condition
evens = [x for x in range(20) if x % 2 == 0]
```

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

```
# Set comprehension
unique_chars = {char for char in "hello world"}
```

String Operations

```
text = "Hello World"
text.lower()      # "hello world"
text.upper()      # "HELLO WORLD"
text.strip()      # Remove whitespace
text.split()      # Split into list
text.replace("Hello", "Hi") # Replace substring
```

```
text.startswith("Hello")    # Boolean check
text.endswith("World")      # Boolean check
len(text)                   # Length
```

File Operations

```
# Reading files
with open("file.txt", "r") as f:
    content = f.read()
    lines = f.readlines()

# Writing files
with open("file.txt", "w") as f:
    f.write("Hello World")
    f.writelines(["Line 1\n", "Line 2\n"])
```

Exception Handling

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero")
except Exception as e:
    print(f"An error occurred: {e}")
else:
    print("No errors occurred")
finally:
    print("This always executes")
```

Classes and Objects

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        return f"Hi, I'm {self.name}"

    def __str__(self):
        return f"Person(name='{self.name}', age={self.age})"
```

```
# Usage
person = Person("Alice", 25)
print(person.greet())
```

Built-in Functions

```
# Math functions
abs(-5)          # Absolute value
min(1, 2, 3)     # Minimum
max(1, 2, 3)     # Maximum
sum([1, 2, 3])   # Sum of iterable
round(3.14159, 2) # Round to 2 decimals

# Type functions
type(42)         # Get type
isinstance(42, int) # Check type
len([1, 2, 3])   # Length

# Iteration functions
enumerate([1, 2, 3]) # Returns (index, value) pairs
zip([1, 2], ['a', 'b']) # Combine iterables
reversed([1, 2, 3]) # Reverse iterator
sorted([3, 1, 2]) # Return sorted list
```

Import and Modules

```
import math
from math import pi, sqrt
import numpy as np
from datetime import datetime, timedelta

# Using imports
math.sqrt(16)
pi
np.array([1, 2, 3])
datetime.now()
```

File and Directory Operations

```
import os
```

```
os.listdir('.')    # List directory contents
os.path.exists('file.txt') # Check if file exists
os.path.join('folder', 'file.txt') # Join paths
```

Date and Time

```
from datetime import datetime, timedelta

now = datetime.now()
tomorrow = now + timedelta(days=1)
formatted = now.strftime('%Y-%m-%d %H:%M:%S')
```

Regular Expressions

```
import re

pattern = r'\d+' # Match digits
re.findall(pattern, "I have 5 apples and 3 oranges")
re.search(pattern, "Age: 25")
re.sub(r'\d+', 'X', "I have 5 apples") # Replace
```