Essential Python Notes for DSA

1. Variables

Explanation:

Variables store data values and allow reuse.

Syntax:

variable_name = value

Example:

x = 10

name = "Alice"

2. Data Types

Explanation:

Data types define the type of data stored in a variable.

Examples:

int: 5

float: 5.5

str: "hello"

list: [1, 2, 3]

3. Type Casting

Explanation:

Convert one data type to another.

Syntax:

int(), float(), str()

Example:

x = int("10")

Essential Python Notes for DSA

4. Operators

Explanation:

Used to perform operations on variables.

Types:

Arithmetic: +, -, *, /, %

Comparison: ==, !=, >, <, >=, <=

Logical: and, or, not

5. Conditional Statements

Explanation:

Used to execute code based on conditions.

Types:

- if
- if-else
- if-elif-else

Syntax:

```
if x > 0:
    print("Positive")
elif x == 0:
    print("Zero")
else:
    print("Negative")
```

6. Loops

Explanation:

Loops repeat a block of code.

Essential Python Notes for DSA

Types:

- for loop

- while loop

Example:

```
for i in range(3):
    print(i)

while i < 5:
    print(i)
    i += 1
```

7. Functions

Explanation:

Functions are reusable blocks of code.

Syntax:

```
def function_name():
    # code
```

Example:

```
def greet():
    print("Hello")
greet()
```

8. Lists

Explanation:

Used to store multiple items.

Syntax:

```
list_name = [item1, item2]
```

Example:

fruits = ["apple", "banana"]

print(fruits[0])

9. Tuples

Explanation:

Like lists, but immutable.

Syntax:

tuple_name = (item1, item2)

Example:

coordinates = (10, 20)

10. Dictionaries

Explanation:

Store key-value pairs.

Syntax:

d = {"key": "value"}

Example:

student = {"name": "John", "age": 20}

11. Sets

Explanation:

Unordered collection of unique items.

Syntax:

s = {1, 2, 3}

Example:

my_set = {1, 2, 2, 3}

print(my_set) # {1, 2, 3}

12. String Manipulation

Explanation:

Strings can be sliced and modified using methods.

Examples:

s = "hello"

print(s.upper())

print(s[1:4])

13. Input/Output

Explanation:

Take input and print output.

Syntax:

input(), print()

Example:

name = input("Enter name:")

print("Hello", name)

14. List Comprehensions

Explanation:

Short way to create lists.

Syntax:

[expression for item in iterable if condition]

Example:

squares = [x*x for x in range(5)]

15. Exception Handling

Explanation:

Used to handle errors safely.

Syntax:

```
try:
    # code
except Exception:
    # handle
```

Example:

```
try:
    x = 1/0
except:
    print("Error")
```

16. Built-in Functions

Explanation:

Common built-in functions used in coding.

Examples:

len(), max(), min(), sum(), sorted()

Example:

print(len([1,2,3]))