**مروری کوتاه بر پایتون**

**تهیه و تنظیم:**

**نیلوفر کاشفی**

**بخش ۱: مبانی پایتون**

برای ورود به دنیای پردازش تصویر، نیاز داریم که ابتدا با مفاهیم پایه‌ای پایتون آشنا شویم. در این جزوه، مباحثی که برای شروع پردازش تصویر ضروری هستند، مرحله‌به‌مرحله توضیح داده شده‌اند.

**۱. نصب پایتون و محیط‌های برنامه‌نویسی**

ابتدا باید پایتون و ابزارهای مورد نیاز را نصب کنیم. اگر پایتون را نصب نکرده‌اید، می‌توانید آن را از [وبسایت رسمی پایتون](https://www.python.org/) دریافت کنید. همچنین می‌توان از محیط‌هایی مانند**Jupyter Notebook** یا**PyCharm** برای برنامه‌نویسی استفاده کرد.

**۲. متغیرها و انواع داده‌ای در پایتون**

در پایتون، متغیرها می‌توانند انواع داده‌ای مختلفی داشته باشند:

x = 10 # عدد صحیح (int)

y = 3.14 # عدد اعشاری (float)

text = "سلام" # رشته متنی (str)

flag = True # مقدار بولی (bool)

**تبدیل انواع داده‌ها**

num = 10

float\_num = float(num) # تبدیل عدد صحیح به اعشاری

str\_num = str(num) # تبدیل عدد به رشته

**۳. عملگرها در پایتون**

**عملگرهای ریاضی**

x = 10

y = 3

print(x + y) # جمع

print(x - y) # تفریق

print(x \* y) # ضرب

print(x / y) # تقسیم

print(x \*\* y) # توان

**عملگرهای مقایسه‌ای**

print(x > y) # بزرگتر

print(x < y) # کوچکتر

print(x == y) # مساوی

**عملگرهای منطقی**

print(True and False) # False

print(True or False) # True

print(not True) # False

**۴. ساختارهای کنترلی: شرط‌ها و حلقه‌ها**

**شرط‌های if-else**

num = 10

if num > 5:

print("عدد بزرگ‌تر از ۵ است")

elif num == 5:

print("عدد برابر با ۵ است")

else:

print("عدد کوچک‌تر از ۵ است")

**حلقه for**

for i in range(5):

print(i)

**حلقه while**

count = 0

while count < 5:

print(count)

count += 1

**بخش ۳: ساختمان داده‌ها در پایتون**

**۱. لیست‌ها**

numbers = [1, 2, 3, 4, 5]

numbers.append(6) # افزودن مقدار جدید

print(numbers[2]) # دسترسی به مقدار ایندکس ۲

**۲. تاپل‌ها (Tuple)**

tuple\_data = (10, 20, 30)

print(tuple\_data[1])

**۳. دیکشنری‌ها**

dict\_data = {"نام": "نیلوفر", "سن": 33}

print(dict\_data["نام"])

**۴. مجموعه‌ها (Set)**

set\_data = {1, 2, 3, 4, 5}

set\_data.add(6)

print(set\_data)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| for | Remove | Insert Item | Existence |  |
| for x in listName  for i in range(len(listName)) | داده مشخص (فقط اولیش):  listName.remove(variable)  اندیس مشخص (خالی:آخری):  listName.pop(index)  حذف کامل:  del listName  حذف متغیرها:  listName.clear() | در اندیس مشخص:  listName.insert(index,variable)  در انتها:  listName.append(variable)  دو تا با هم:  listName1.extend(listName2)  listName1+listName2 | Variable in listName | List |
| for x in tupleName  for i in range(len(tupleName)) |  | tupleName1+tupleName2 | Variable in tupleName | Tuple |
| for x in setName  for i in range(len(setName)) | داده مشخص:  setName.remove(variable)  رندوم:  setName.pop()  حذف کامل:  del setName  حذف متغیرها:  setName.clear() | یک عنصر:  setName.add(variable)  دو تا با هم:  setName.update(object2) | Variable in tupleName | Set |
| for x in dictName  for i in range(len(dictName)) | داده مشخص:  dictName.pop(variable)  آخری:  setName.popitem()  حذف کامل:  del setName  حذف متغیرها:  setName.clear() | dictName[“new index”]=var | Variable in dictName | Dictionary |

**بخش ۴: توابع و برنامه‌نویسی ماژولار**

**۱. تعریف تابع**

def greet(name):

return "سلام " + name

print(greet("نیلوفر"))

**۲. استفاده از توابع کتابخانه‌ای**

import math

print(math.sqrt(25)) # محاسبه جذر عدد ۲۵

**۳. کار با فایل‌ها**

# نوشتن در فایل

with open("sample.txt", "w") as file:

file.write("این یک تست است")

# خواندن از فایل

with open("sample.txt", "r") as file:

content = file.read()

print(content)

**بخش ۵: کار با آرایه‌ها و ماتریس‌ها با NumPy**

**۱. تعریف آرایه‌های NumPy**

import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

**۲. عملیات پایه روی آرایه‌ها**

arr2 = arr \* 2 # ضرب همه مقادیر در ۲

mean\_value = np.mean(arr) # میانگین مقادیر آرایه

**۳. کار با ماتریس‌ها**

matrix = np.array([[1, 2], [3, 4]])

print(matrix.shape) # نمایش ابعاد ماتریس

**بخش 6: خلاصه**

|  |  |
| --- | --- |
| **Case sensitive** | Both parameters and keyword |
| **Blocks** | No {}, No ;, Using tab |
| **Comment** | # This is also a single-line comment  """ This is a comment written in  more than just one line """ |
| **Declaration** | x = 5 No declare |
| **Assign together identical values** | x = y = z = "Orange" |
| **Assign together different values** | x, y, z = "Orange", "Banana", "Cherry"  fruits = ["apple", "banana", "cherry"] x, y, z = fruits |
| **Print**  **Type** | x = 5  x = ("apple", "banana", "cherry")  print(type(x)) |
| **Casting** | z = float(3)  # z will be 3.0 |
| **Output variable** | w= 5  x = "Python " y = "is " z = "awesome" print(x + y + z) # Python is awesome  print(x, y, z) # Python is awesome  print(w + z) # Error  print(w, z) # 5 awesome |
| **Global Variables** | Variables that are created outside of a function can be used inside the function, but they can’t change. Inside the function: **Use “global” keyword**  **To: change/use outside the function** |
| **String Dec.** | "John" = 'John' |
| **Sum: str and int** | x = 5 y = "4John" print(x+y) # Error |
| **Operators** | **Logical Operators:** And / or / not**Identity Operators (same object):** is / is not **Comparison Operators (same value): == / !=** **Membership Operators:** in – not in**Bitwise Operators:** & | ~(not) ^(XOR) |
| **Strings are Arrays** | **Access:** a[i]  **Loop:** for x in "banana":  **Length:** len(a)  **Search:**  txt = "The best things in life are free!" if "free" in txt:  if "expensive" not in txt:  **Slice: a[2:5] a[:5] a[2:] a[-1]** # End |
| **String methods** | a = "Hello, World!"  a.upper() / a.lower()  print(a.strip()) # returns "Hello,World!"  print(a.replace("H", "J")) #returns "jello,World!"  print(a.split(",")) # returns ['Hello', ' World!']  a.count("o") # returns 2 |
| **String Concatenation** | a = "Hello" b = "World" c = a + b |
| **Boolean** | **True values:** any string, any number, any datatype with value, yes, true  **False values:** 0, none, empty, false, no |
| **built-in data types** | **List:** ordered, changeable, Allow Duplicates, contains different data types**Tuple:** ordered, unchangeable, Allows Duplicates, contains different data types**Set:** unordered, unchangeable, No Duplicates, contains different data types**Dictionary:** ordered, changeable, No Duplicates, contains different data types |
| **built-in data types methods** | a.insert(index, value)  a.append(value) # to the end  a.extend(b) # append to data type  b=a.copy() # b=a is false(b will be reference)  a.remove(value) # specific item  a.pop(i) / del a[i] # specific index  del a # delete a entirely  a.clear() # just clear |
| **loop** | fruits = ["apple", "banana", "cherry"] for x in fruits:   print(x)  for x in range(2, 30, 3):   print(x)  while i < 6:   print(i)   i += 1 |
| **condition** | a = 200 b = 33 if b > a:   print("b is greater than a") elif a == b:   print("a and b are equal") else:   print("a is greater than b") |