

ПРАКТИКА 2

Завдання 1

1. Напишіть скрипт, де у випадку якщо значення змінної більше нуля, то друкується повідомлення (використати функцію print). В іншому випадку виконати операцію підсумовування.	<pre>n = int(input("Введіть число: ")) if n>0: print("Введенное число больше нуля") else: m = n + n print("Введенное число меньше нуля, операция суммирования n + n: ", m)</pre>
2. Змініть коди на основі else так, щоб залежно від значень змінної друкувалось 1 або -1.	<pre>n = int(input("Введіть число: ")) if n>0: print("1") else: print("-1")</pre>
3. Сформуйте програму з if та else, де "вкладений код" повинен мати не менше 3 виразів.	<pre>var1 = int(input("Введіть первое число: ")) var2 = int(input("Введіть второе число: ")) var3 = int(input("Введіть третье число: ")) if var3>var1: der = var3-var1 der = pow(der,2) print ("var3 больше var1, их разница в квадрате: ", der) else: sum = var3+var1-var2 sum = pow(sum,2) print ("var3 меньше var1, значение var3 + var1 - var2 в квадрате: ", sum) print ("Good bye! Have a good day!")</pre>

Завдання 2

1. Створіть програму за наступним алгоритмом: А) двом змінним присвоїти певні числові значення на вибір (не нульові); Б) якщо значення першої змінної більше за другу, то знайти різницю значень змінних і результат присвоїти третій змінній. У протилежному випадку, третя змінна повинна дорівнювати сумі 2-х перших змінних. В) У всіх інших випадках присвоїти третій змінній значення першої змінної. Г) Вивести на екран значення 3-ї змінної.	<pre>var1 = int(input("Введіть первое число отличное от нуля: ")) var2 = int(input("Введіть второе число отличное от нуля: ")) if var1>var2: var3 = var1 - var2 elif var1<var2: var3 = var1 + var2 else: var3 = var1 print ("Значение третьей переменной: ", var3)</pre>
2. Складіть програму з оператором if-elif-else, де кількість ланок щонайменше 4.	<pre>num1 = int(input("Введіть x: ")) if num1 == 0: result=0 elif num1==1: result=1 elif num1==2: result=2 elif num1==3: result=3 elif num1==4: result=4 elif num1==5: result=5 else: print ("Error") result = 0 print ("Result: ", result)</pre>

Завдання 3

1. Написати скрипт виведення чисел Фібоначі (до 100).	<pre>fib1 = 0 fib2 = 1 print (fib1) print (fib2) n = 10 i = 0 fib_sum = 0 while i < n and fib_sum<100: fib_sum = fib1 + fib2 print (fib_sum) fib1 = fib2 fib2 = fib_sum i = i + 1</pre>
2. Змінити код так, щоб виведення ряду починалось з 5 члена і закінчувалось 20 членом ряду.	<pre>fib5 = 3 fib6 = 5 print (fib5) print (fib6) n = 18 i = 5 fib_sum = 0 while i < n: fib_sum = fib5 + fib6 print (fib_sum) fib5 = fib6 fib6 = fib_sum i = i + 1</pre>
3. Написати програму виведення парних чисел від 0 до 20 і потім вивести кожне третє число в ряді від -1 до -21.	<pre>print ("Парные числа: ") for x in range (0,21): if x !=0 and x % 2 == 0: print (x) print ("Каждое третье число: ") for i in range(-1,-22,-2): if i !=-1: print(i)</pre>

Завдання 4

1. Створіть скрипт (data.py), який би питав користувача його ім'я, вік, місце проживання. А потім выводив би 3 рядки: There is ім'я; It is вік; He live in місце проживання. Тут замість вкладень повинні бути дані, які введені користувачем.	<pre>userName = input('Введите Ваше имя: ') userAge = input('Введите Ваш возраст: ') userAdress = input('Введите место Вашего проживания: ') print("This is", userName) print("It is", userAge) print("He live in", userAdress)</pre>
2. Напишіть програму (example.py) яка б пропонувала користувачу розв'язати приклад $5*58-27$. Якщо користувач пише правильну відповідь, то вітання від програми, якщо ні – помилка (використати if-else).	<pre>answ=int(input("Решите пример 5*58-27: ")) if answ == 263: print("ПРАВИЛЬНО!") else: print("НЕПРАВИЛЬНО!")</pre>
3. Перепишіть програму так, щоб користувач розв'язував приклад, до тих пір, поки він не напише правильну відповідь (7 спроб).	<pre>print ("У вас есть 7 попыток") for i in range(0,7,1): answ=int(input("Решите пример 5*58-27: ")) if answ == 263: print("ПРАВИЛЬНО!") else: print("НЕПРАВИЛЬНО!")</pre>

ПРАКТИКА 3

Завдання 1

1. Визначте рядок не менше з 8 символів і виведіть на екран з цього рядка: перший символ, останній символ, третій з початку та третій з кінця рядка.	<pre>print("Введіть рядок не менше ніж з 8 символів взявши його в подвійні лапки: ") s=str(input()) length=len(s) first=s[0] last=s[-1] thstart=s[2] thend=s[-3] print print ("Перший символ: " + first) print ("Останній символ: " + last) print ("Третій символ з початку: " + thstart) print ("Третій символ з кінця: " + thend)</pre>
--	---

Завдання 2

1. Створіть 2 списки і присвоїти їм змінні. Створіть 3 список, де поєднано елементи послідовно з 2 списків: 1212 і вивести її на екран.	<pre>print("Введіть кількість елементів першого та другого списків: ") k=int(input()) list1 = [] list2 = [] list3 = [] i=0 j=0 k3=2*k l=0 print ("Введіть елементи першого списку натискаючи enter після кожного") for i in range(k): new_element = int(input()) list1.append(new_element) print ("Введіть елементи другого списку натискаючи enter після кожного") for j in range(k): new_element2 = int(input()) list2.append(new_element2) print ("Ви створили два списки") print list1 print list2 print print ("Список, де поєднано елементи послідовно з 2 списків: 1212") i=0 j=0 for l in range(0, k3, 1): if l%2!=0 and l!=0: list3.append(list2[j]) j=j+1 else: list3.append(list1[i]) i=i+1 print list3</pre>
---	--

Завдання 3

1. Створити список з 5 чисел і кожне з них збільшити у 4 рази.	<pre>list4 = [] p=5 o=0 print ("Введіть 5 елементів списку натискаючи enter після кожного") for o in range(p): new_element3 = int(input()) list4.append(new_element3) print ("Ви створили список") print list4 print ("Список з числами збільшеними у 3 рази") for o in range(p): list4[o]=list4[o]*3 print list4</pre>
--	---

Завдання 4

1. Створити список з 4 рядків. За допомогою циклу for вивести ці рядки наскран.	<pre>list4 = ["good morning", "faculty of electronics", "i hope i pass exam", "i love new year"] p=0 print ("Ви створили наступний список:") print (list4) print () print ("Виведення елементів списків за допомогою циклу for: ") for p in range(0,4,1): print (list4[p])</pre>
2. Змінити код так, щоб в кінці кожної літери додавалась крапка.	<pre>list5 = ["today", "tomorrow", "yesterday", "KPI"] u=0 f=4 m=0 print ("Створений список: ") for u in range(f): print (list5[u]) print ("Список з рядками, де в кінці кожної літери додається крапка") for u in list5: for m in u: end="." print(m+end) ЕСЛИ НАДО В КОНЦЕ КАЖДОГО СЛОВА, ТО НАПИСАТЬ ЦИКЛ for u in list5: end="." print(u+end)</pre>
3. Створіть список з чисел і змінити тип даних на числа з плаваючою точкою.	<pre>list6 = [15,79,43,78] list7 = [] print ("Створений список з чисел") print (list6) i = 1 for a in range(4): t= list6[a] list6[a] = float(t) print ("Список з чисел після зміни типу даних на числа з плаваючою крапкою") print (list6)</pre>

ПРАКТИКА 4

Завдання 1

1. Написати код, з використанням функцій, який б з поверненням до основної програми зміг би розрахувати вираз: $(a+b)-c \cdot 5$ (d)	<pre>def func(): a=5 b=10 c=-2 d=20 answ=(a+b)-(pow(c,5))*d return answ print(func())</pre> <p>ЕСЛИ (d) ИМЕЕТСЯ В ВИДУ МОДУЛЬ ЧИСЛА ТО Надо імпортировать модуль в самом начале: <code>import math</code> Поменять 6-ю строку: <code>answ=(a+b)-(pow(c,5))*math.fabs(d)</code></p>
--	--

Завдання 2

1. Написати код, з використанням функцій та аргументів, який б з поверненням до основної програми зміг би розрахувати вираз: $(a+b)-c \cdot 5$ (d)	<pre>def func(a,b,c,d): answ=(a+b)-(pow(c,5))*d return answ print(func(5,10,-2,20))</pre> <p>ЕСЛИ (d) ИМЕЕТСЯ В ВИДУ МОДУЛЬ ЧИСЛА ТО Надо імпортировать модуль в самом начале: <code>import math</code> Поменять 2-ю строку: <code>answ=(a+b)-(pow(c,5))*math.fabs(d)</code></p>
--	--

Завдання 3

1. Створити функцію <code>def fcl(num): n=num:10 print(n)</code> Викликати цю функцію, де в якості аргументів – глобальна змінна, число, рядок.	<pre>globalvar=10 def fcl(globalvar, num, str): n=num/10 print("Ответ: ") print(n) print('Globalvar: ') print(globalvar) print('String: ') print(str) new=fcl(globalvar, 10, "practice4")</pre>
--	---

Завдання 4

1. Створити генератор паролей з цифр 0...9.	<pre>import random Num_S=list('1234567890') ln=4 //задаем количество цифр в пароле Pass="" PassA="" i = 0 def Num(): for i in range(ln): global Pass n=random.randrange(len(Num_S)) Pass=Pass+Num_S[n] return Pass def All(): global PassA if ln>0: PassA=Num() PassR=list(PassA) random.shuffle(PassR) PassEnd="" for i in range(len(PassR)): PassEnd=PassEnd+PassR[i] return PassEnd print (All())</pre>
2. Зашифрувати рядок у формі SHA512.	<pre>import hashlib h=hashlib.sha512(b'Good morning, beautiful world') answ=h.hexdigest() print(answ)</pre> <p>ЕСЛИ ДРУГОЙ АЛГОРИТМ ШИФРОВАНИЯ, ТО ВО 2-Й СТРОКЕ СТАВИМ: <code>sha1(), sha224(), sha256(), sha384(), sha512(), blake2b(), blake2s(), md5()</code></p>

ПРАКТИКА 5

Завдання 1

<p>1. Написати код, виведення дати за машинним годинником системи.</p>	<pre>import datetime today = datetime.datetime.today() print(today.strftime("%d/%m/%Y"))</pre> <p>ЕСЛИ НАДО ДАТУ/ВРЕМЯ РАЗНЫМИ МЕТОДАМИ (НЕ ЗАБЫВАТЬ В НАЧАЛЕ ПОДКЛЮЧАТЬ МОДУЛЬ datetime) и везде с print СКОБКИ!!</p> <pre>now = datetime.datetime.now() print "Текущая дата и время с использованием метода str:" print str(now) print "Текущая дата и время с использованием атрибутов:" print "Текущий год: %d" % now.year print "Текущий месяц: %d" % now.month print "Текущий день: %d" % now.day print "Текущий час: %d" % now.hour print "Текущая минута: %d" % now.minute print "Текущая секунда: %d" % now.second print "Текущая микросекунда: %d" % now.microsecond print "Текущая дата и время с использованием strftime:" print now.strftime("%d-%m-%Y %H:%M") print "Текущая дата и время с использованием isoformat:" print now.isoformat()</pre>
<p>2. Написати код перевірки наявності у модулі math двох ідентифікаторів- змінних – число Пі та змінна у.</p>	<pre>import math lst=(dir (math)) n=len(lst) x=(lst.count('pi')) y=(lst.count('y')) if x==1: print("В модуле есть индентификатор числа pi") else: print("В модуле нет индентификатора числа pi") if y==1: print("В модуле есть индентификатор переменной y") else: print("В модуле нет индентификатора переменной y")</pre>

Завдання 2

<p>1. Зберегти на диску зображення та завантажити його у середовище програмування</p>	<pre>from PIL import Image, ImageDraw image = Image.open("img1.jpg") //дальше в ручную меняем цвет пикселей по координатам image.putpixel((25,45),(255,0,0)) image.putpixel((30,45),(255,0,0)) image.putpixel((35,45),(255,0,0)) image.putpixel((40,45),(255,0,0)) image.putpixel((45,45),(255,0,0)) image.putpixel((50,45),(255,0,0)) image.putpixel((55,45),(255,0,0)) image.putpixel((60,45),(255,0,0)) image.putpixel((65,45),(255,0,0)) image.putpixel((70,45),(255,0,0)) image.save("img2.jpg","JPEG",quality=95)</pre>
<p>2. У зображенні змінити 10 пікселів.</p>	
<p>3. Зберегти змінене зображення з якістю 95%.</p>	

Завдання 3

<p>1. Створити зображення – синій прямокутник з розмірами 200*1000 пікселів і забезпечити у ньому канал прозорості.</p>	<pre>from PIL import Image, ImageDraw //тут размеры могут быть перепутаны, т.е. (200, 1000) image = Image.new("RGBA", (1000,200), (0,0,0,0)) //создаем область для рисования draw = ImageDraw.Draw(image) //с помощью рисования квадрата, рисуем прямоугольник через правильно заданные координаты вершин draw.rectangle((0,0,1000,200),fill="blue",outline="blue") //удаляем область для рисования del draw image.save("image1.jpg","JPEG")</pre>
---	--

ПРАКТИКА 6

Завдання 1

1. Для трьох зображень зменшити їх розміри у 4 рази пропорційно через фільтри NEAREST, BICUBIC, BILINEAR. Порівняти отримані зображення за якістю.	<pre> from PIL import Image //открываем 4 картинки Img1 = Image.open("img1.jpg") Img2 = Image.open("img2.jpg") Img3 = Image.open("img3.jpg") Img4 = Image.open("img4.jpg") //узнаем размер Img1.size Img2.size Img3.size Img4.size //меняем размер через разные фильтры Img1.thumbnail((size,size),Image.NEAREST) Img2.thumbnail((size,size),Image.BICUBIC) Img3.thumbnail((size,size),Image.BILINEAR) //сохраняем Img1.save('im1.jpg') Img2.save('im2.jpg') Img3.save('im3.jpg') </pre>
2. Змінити розмір 4 зображення без залучення алгоритмів модуля Image.	<pre> С ПОМОЩЬЮ МОДУЛЯ resizeimage: from resizeimage import resizeimage I4 = resizeimage.resize_thumbnail(Img4, size) I4.save('im4.jpg') С ПОМОЩЬЮ МОДУЛЯ Image: from PIL import Image I41 = Img4.thumbnail(size) I41.save('im41.jpg') </pre>
3. Друге вихідне зображення повернути на 145 градусів проти годинникової стрілки.	<pre> //поворот + сохраняем под новым именем Img5 = Img2.rotate(145) Img5.save('im2r.jpg') </pre>
4. З третім зображенням виконати процедуру віддзеркалювання елементів зверху до низу.	<pre> // отражение элементов сверху вниз с третьим выходным изображением Img6 = Img3.transpose(Image.FLIP_TOP_BOTTOM) </pre>

Завдання 2

1. Застосувати до обраного зображення 2 фільтри з модуля ImageFilter SMOOTH, SHARPEN, BLUR. Порівняти зображення між собою.	<pre> //ИМПОРТ Image И ImageFilter from PIL import Image, ImageFilter //ИМПОРТ НЕОБХОДИМЫХ ФИЛЬТРОВ from PIL.ImageFilter import (BLUR, SMOOTH, SHARPEN) //открываем новое изображение origin = Image.open("image1.jpg") //применяем фильтры blurred = origin.filter(ImageFilter.BLUR) sharpen = origin.filter(ImageFilter.SHARPEN) smoothen = origin.filter(ImageFilter.SMOOTH) //сохраняем blurred.save('blurred.jpg') sharpen.save('sharpen.jpg') smoothen.save('smoothen.jpg') СРАВНЕНИЕ! Фильтр SHARPEN - самый четкий, но очень зернистый. Фильтр SMOOTH слегка размыт. BLUR - очень размыто. </pre>
2. Створити скриншот екрану розміром 300 на 300 пікселів і перевірити результат збережених даних.	<pre> //ИМПОРТ ImageGrab для скриншота from PIL import Image, ImageGrab //создаем и сохраняем скриншот screen = ImageGrab.grab((300, 300)) screen.save('screen.bmp', 'BMP') //проверяем размеры screen.size </pre>

Для NUKE

nuke.message('hello')

Для відображення вікна введення інформації (крапка і дужки обов'язкові елементи)

nuke.getInput('заголовок','текст по умовчанию')

Вікно виклику файлу можна викликати через команду

nuke.getFilename('test')

Додати ноду

nuke.nodes.Blur()

нову ноду з'єднати з виділеною і створеною вже раніше у вікні NodeGraph автоматично

nuke.createNode('Blur')

без вікна властивостей

nuke.createNode('Blur', inpanel=False)

зміна параметра ноди

nuke.nodes.Blur(size=10)

створення нової з параметром

nuke.createNode('Blur', 'size 10')

Присвоїти змінну

b=nuke.nodes.Blur()

Вивести значення параметру

print b['size'].value()

анімація параметру ноди

b1=nuke.nodes.Blur()

v=b1['size']

v.setAnimated()

на першому кадрі

v.setValue(10, time=1)

новий параметр

k=nuke.Array_Knob('Param', 'new_param')

де Array – тип комірки; (WH- бігунок, слайдер; Boolean – позначка у вікні властивостей за параметром) і ми лише в змінну внесли деякі параметри. Для включення цього параметра у ноду, наприклад, Blur **b.addKnob(k)**

Місцезнаходження доданої ноди Grade в панелі NodeGraph:

g=nuke.nodes.Grade(xpos=0, ypos=0) або **b.setXpos(50)**

З'єднання нодів (два входи)

```
m=nuke.nodes.Merge(inputs=[g,b])
```

підключити стару б1 до м

```
b1.setInput( 0, m )
```

виділити ноду під назвою Transform1 `nuke.toNode('Transform1').setSelected(True)`

імя виділеної ноди

```
nuke.selectedNode().name()
```

виділити всі ноди

```
nuke.selectedNodes()
```

додати файл

```
r=nuke.nodes.Read(file='C:/Temp/1.png')
```

підключення до екрану

```
nuke.connectViewer( 0, m )
```

збереження файлу

```
w=nuke.nodes.Write(file='C:/Temp/test.####.jpg')
```

рендерінг 1-50 кадрів

```
Nuke.render('Write2', 1, 50)
```

ПРОСТО ПИТОН

Функции if, else, elif:

if a > b:

c=a-b

print(c)

elif a<b:

c=a+b

print(c)

else:

print('Error')