Моя IT Школа

сертифицированные IT курсы

Основы языка Python

План занятия

Моя IT Школа
сертифицированные IT курсы

1. PEP8
2. Встроенные функции
3. Арифметические операции с числами
4. Работа с модулем math
5. Работа с модулем random

PEP8

Этот документ описывает соглашение о том, как писать код для языка python, включая стандартную библиотеку, входящую в состав python.

PEP 8 создан на основе рекомендаций Гвидо Ван Россума с добавлениями от Барри. Если где-то возникал конфликт, мы выбирали стиль Гвидо. И, конечно, этот PEP может быть неполным (фактически, он, наверное, никогда не будет закончен).

Ключевая идея Гвидо такова: код читается намного больше раз, чем пишется. Собственно, рекомендации о стиле написания кода направлены на то, чтобы улучшить читаемость кода и сделать его согласованным между большим числом проектов. В идеале, весь код будет написан в едином стиле, и любой сможет легко его прочесть.

Это руководство о согласованности и единстве. Согласованность с этим руководством очень важна. Согласованность внутри одного проекта еще важнее. А согласованность внутри модуля или функции — самое важное. Но важно помнить, что иногда это руководство неприменимо, и понимать, когда можно отойти от рекомендаций. Когда вы сомневаетесь, просто посмотрите на другие примеры и решите, какой выглядит лучше.

Две причины для того, чтобы нарушить данные правила:

1. Когда применение правила сделает код менее читаемым даже для того, кто привык читать код, который следует правилам.

1. Чтобы писать в едином стиле с кодом, который уже есть в проекте и который нарушает правила (возможно, в силу исторических причин) — впрочем, это возможность переписать чужой код.

Встроенные функции

Моя IT Школа

В Python существуют десятки встроенных функций и классов, сотни инструментов, входящих в стандартную библиотеку Python, и тысячи сторонних библиотек на PyPI.

Чтобы разобраться, на какие функции стоит обратить внимание, их следует разделить на группы:

- общеизвестные: почти все новички используют эти функции и довольно часто;
- неочевидные для новичков: об этих функциях полезно знать, но их легко пропустить, когда вы новичок в Python;
- понадобятся позже: об этих встроенных функциях полезно помнить, чтобы найти их потом, когда/если они понадобятся;
- можно изучить когда-нибудь: это может пригодиться, но только при определённых обстоятельствах;
- скорее всего, они вам не нужны: они вряд ли понадобятся, если вы не занимаетесь чем-то достаточно специализированным.

Преобразование типов

Моя IT Школа

type — вернет тип объекта;
str — преобразование в строку;
int — преобразование в число;
float — преобразование в число с плавающей точкой;
complex — преобразование в комплексное число;
bool — преобразование к булевому типу;
tuple — преобразование к кортежу;
dict — преобразование к словарю;
frozenset — приведение к неизменяемому множеству;
list — приведение к списку;
set — преобразование к множеству;
slice — создание среза;
bin — приведение целого числа к двоичной строке;
hex — целое число в шестнадцатеричную строку;
oct — целое число в восьмеричную строку.

Работа с числами

Моя IT Школа

В Python достаточно просто работать с числами, ведь сам язык является простым и одновременно мощным. Он поддерживает всего три числовых типа:

- **int (целые числа)**
- **float (числа с плавающей точкой)**
- **complex (комплексные числа)**

Арифметические операции над числами

Моя IT Школа

Оператор в Python — это символ, который выполняет операцию над одним или несколькими операндами.
Операндом выступает переменная или значение, над которыми проводится операция.

Математические операторы

Оператор	Описание	Пример	Результат
+	Сложение	7 + 3	10
-	Вычитание	7 - 3	4
*	Умножение	7 * 3	21
/	Деление (истинное)	7 / 3	2.3333333333333335
**	Возведение в степень	7**3	343
//	Целочисленное деление	7 // 3	2
%	Остаток от деления	7 % 3	1

Порядок операций:

1. Скобки
2. Возведение в степень
3. Умножение
4. Деление
5. Сложение
6. Вычитание

Напишем программу, которая выполняет простые арифметические операции

```
a = 6
b = 2

# операция сложения
d = a + b
# операция вычитания
f = a - b

# напишите далеее сами
```

```
# напишите далеее сами

# операция деления
g = a/b
# остаток от деления
h = a%b
# возведение в степень
t = a**b
# деление нацело
w = a//b

print(d)
print(f)
print(g)
print(h)
print(t)
print(w)
```

Результат работы 2-ой программы:

```
8
4
3.0
0
36
3
```

Задание №1

Ввести 3 числа и выполнить с ними все математические операции в различных комбинациях(+,-,/,)**

Решение

```
a = int(input("Введите число: "))
b = int(input("Введите число: "))
c = int(input("Введите число: "))

print("Сумма: ", a + b, "разность: ", a - b, "произведение: ", a * b, "деление: ", a / b, "возведение в степень: ", a **b)
print("Сумма: ", a + c, "разность: ", a - c, "произведение: ", a * c, "деление: ", a / c, "возведение в степень: ", a **c)
print("Сумма: ", b + c, "разность: ", b - c, "произведение: ", b * c, "деление: ", b / c, "возведение в степень: ", b **c)
print("Разность: ", b - a, "деление: ", b / a, "возведение в степень: ", b ** a)
print("Разность: ", c - a, "деление: ", c / a, "возведение в степень: ", c ** a)
print("Разность: ", c - b, "деление: ", c / b, "возведение в степень: ", c ** b)

print("Сумма: ", a + b + c, "разность: ", a - b - c, "произведение: ", a * b * c, "деление: ", a / b / c)
print("Разность: ", b - a - c, "деление: ", b / a / c)
print("Разность: ", c - a - b, "деление: ", c / a / b)
```

Задание №2

Найти значения выражений:
17/2*3+2
2+17/2*3
19%4+15/2*3
(15+6)-10*4
17/2%2*3**3

Решение

```
2  print(17 / 2 * 3 + 2)
3  print(2 + 17 / 2 * 3)
4  print(19 % 4 + 15 / 2 * 3)
5  print((15 + 6) - 10 * 4)
6  print(17 / 2 % 2 - 3 ** 3)
```

Задание №3

Анна пошла в магазин, у нее было 11 рублей. Хлеб стоит 1 рубль 50 копеек. Анна купила 3 буханки хлеба. Сколько рублей у неё осталось?

Решение

```
24  rubles = 11
25  bread = 1.5
26  last = 11 - 1.5 * 3
27  print(last)
```

Задание №4

Найдите средний возраст студентов вашей группы

Модуль math

Моя IT Школа

Библиотека Math в Python обеспечивает доступ к некоторым популярным математическим функциям и константам, которые можно использовать в коде для более сложных математических вычислений. Библиотека является встроенным модулем Python.

Чтобы подключить модуль, необходимо в начале программы прописать следующую инструкцию:

```
1  import math
2
```

math.ceil(X) – округление до ближайшего большего числа.
math.factorial(X) - факториал числа X.
math.floor(X) - округление вниз.
math.fmod(X, Y) - остаток от деления X на Y.
math.isfinite(X) - является ли X числом.
math.isnan(X) - является ли X NaN (Not a Number - не число).
math.trunc(X) - усекает значение X до целого.
math.exp(X) - eX.
math.log1p(X) - натуральный логарифм (1 + X). При X → 0 точнее, чем math.log(1+X).

math.log10(X) - логарифм X по основанию 10.
math.log2(X) - логарифм X по основанию 2.
math.pow(X, Y) - XY.
math.sqrt(X) - квадратный корень из X.
math.cos(X) - косинус X (X указывается в радианах).
math.sin(X) - синус X (X указывается в радианах).
math.tan(X) - тангенс X (X указывается в радианах).
math.hypot(X, Y) - вычисляет гипотенузу треугольника с катетами X и Y (math.sqrt(x * x + y * y)).
math.degrees(X) - конвертирует радианы в градусы.
math.radians(X) - конвертирует градусы в радианы.
math.pi - pi = 3,1415926...
math.e - e = 2,718281...

Задание №5

Напишите программу для преобразования градусов в радианы.

Примечание. Радиан - это стандартная единица измерения углов, используемая во многих областях математики. Измерение угла в радианах численно равно длине соответствующей дуги единичного круга; один радиан чуть меньше 57,3 градуса (когда длина дуги равна радиусу).

Решение

```
20  pi = 22 / 7
21  degree = float(input("Введите градусы: "))
22  radian = degree * (pi / 180)
23  print(radian)
24
```

```
25  degree = float(input('Введите градусы: '))
26  radians = math.radians(degree)
27  print(radians)
28
```

Задание №6

Напишите программу для вычисления значения дискриминанта.

Примечание: дискриминант - это имя, данное выражению, которое появляется под знаком квадратного корня (радикала) в квадратной формуле.

Решение

```
20  x = float(input('Значение x: '))
21  y = float(input('Значение y: '))
22  z = float(input('Значение z: '))
23  discriminant = (y**2) - (4*x*z)
24
```

Задание №7

Найти площадь и периметр прямоугольного треугольника

Решение

```python
# Импортируется модуль math, содержащий
# различные математические функции.
import math

# функция input() возвращает строку
AB = input("Длина первого катета: ")
AC = input("Длина второго катета: ")

# строки переводятся в вещественные числа
AB = float(AB)
AC = float(AC)

# Находим гипотенузу по теореме Пифагора:
# "сумма квадратов катетов
# равна квадрату гипотенузы".
# Функция sqrt() из модуля math
# извлекает квадратный корень.
# Оператор  возводит в квадрат.
BC = math.sqrt(AB ** 2 + AC ** 2)

# Площадь прямоугольного треугольника
# равна половине площади
# соответствующего прямоугольника.
S = (AB * AC) / 2

# Периметр находится как сумма всех сторон.
P = AB + AC + BC
```

Модуль random

Моя IT Школа

Случайные числа применяются в программировании в разных случаях, например, для моделирования процессов и в видеоиграх. Для начала разберёмся, какую последовательность можно назвать случайной.

Случайной последовательностью называют набор элементов, полученных таким образом, что любой элемент их этого набора никак не связан ни с каким другим элементом. При этом в программировании обычно последовательность не является строго случайной — в ней для генерации следующего элемента используется предыдущий.

В модуле random реализованы различные генераторы псевдослучайных чисел. Здесь присутствуют методы, с помощью которых можно получить равномерное, Гауссовское, бета и гамма распределения и другие функции. Практически все они зависят от метода random(). В Python, в качестве основного, используется генератор псевдослучайных чисел Mersenne Twister, который выдает 53-х битные вещественные числа.

Чтобы начать использовать встроенные генераторы случайных чисел, нужно сначала подключить модуль:

```
1    import random
```

Основные функции модуля random

Моя IT Школа

random.randint(A, B) - случайное целое число N, A ≤ N ≤ B.

random.uniform(A, B) - случайное число с плавающей точкой, A ≤ N ≤ B.

random.random() - случайное число от 0 до 1.

random.choice(sequence) - случайный элемент непустой последовательности.

Задание №8

Вычислить сумму цифр случайного трёхзначного числа

Решение

```python
import random

# При умножении на 900 получается случайное число от 0 до 899.(9).
# Если прибавить 100, то получится число от 100 до 999.(9).

n = random.randint(100,999)
print(n)

# Извлекается первая цифра (старший разряд) числа путем
# деления нацело на 100

a = n // 100

# Деление нацело на 10 удаляет последнюю цифру числа.
# Затем нахождение остатка при делении на 10 извлекает
# последнюю цифру,которая в исходном числе была средней.

b = (n // 10) % 10

# Последняя цифра (младший разряд) числа находится
# путем нахождения остатка при делении нацело на 10.

c = n % 10

# Вычисляется сумма цифр и выводится на экран

print(a + b + c)
```

Творческое задание

Придумайте задачу на пройденные 2 темы и решите её

Домашнее задание

Моя IT Школа

Задача №1

У Анны 2 яблока, у Пола 5 яблок. Выведите на экран сколько яблок у Пола и сколько яблок у Анны одной командой print. Использовать переменные

Задача №2

Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности

Задача №3 (задача на логику)

Улитка в день проползает 2м, но за ночь сползает на 1 м. За сколько дней улитка доползает, если дерево в высоту 20м