

Тема: Логический тип данных (Bool).

Понятие.

1. Переменные логического типа могут принимать только одно из двух значений **True** (истина), **False** (ложь).

$X \% 2 == 0$ - результат выполнения операции bool.

$X = 10 > 0$ - результат присваивания, значение логической переменной (bool)

Туре (x) – выдает bool.

Примеры использования:

- 1) В командной строке можно указывать выражение $5 > 0$; $12 * 3 \% 2 == 0$; `'кот' == 'кит'`

- 2)

```
b = int(input())
print(b % 2 == 0)
```

- 3)

```
f = True
if f:
    print('Да')
else:
    print('Нет')
```

В данном примере нет выражения, переменной `f` присвоено значение `True`. Оператор ветвления передает управление тому действию, которое соответствует этой переменной. Подобные приемы часто используют в виде «флажка» (да или нет).

2. **Сложное ветвление.** Часто в задачах нужно реализовать не две ветки, а три, четыре и т.д. В этом случае есть два пути.

Первый – разветвлять внутри оператора ветвления:

```
if a > b:
    print("Первое число больше второго")
else:
    if a == b:
        print('числа равны')
    else:
        print('Второе число больше первого')
```

В этом примере реализованы три ветки, так можно делать и более сложные. Но это не совсем удобно и можно запутаться. В Python есть оператор множественного ветвления:

```
if условие 1:
    действие1
elif условие 2:
    действие2
elif условие 3:
```

```

        действие3
elif условие 4:
    действие1
....
else:
    Действие, если ни одно условие не выполняется.

```

Задача1. «Калькулятор». Напишите программу, которая вводит два число и выполняет то или иное арифметическое действие

```

f_num = int(input("Введите число"))
oper = input("Введите операцию")
s_num = int(input("Введите второе число"))
if oper == "+":
    print(f_num + s_num)
elif oper == "-":
    print(f_num - s_num)
elif oper == "*":
    print(f_num * s_num)
elif oper == "/":
    print(f_num / s_num)
else :
    print("Нет такой операции")

```

Рассмотрите самостоятельно следующую задачу.

Задача 2. _Китайский гороскоп делит время на 12-летние циклы, и каждому году соответствует конкретное животное. Год, который делится на 12 – год Обезьяны, остальные идут по порядку:

Год	Животное
2004	Обезьяна
2005	Петух
2006	Собака
2007	Свинья
2008	Крыса
2009	Бык
2010	Тигр
2011	Кролик
2012	Дракон
2013	Змея
2014	Лошадь
2015	Коза

Напишите программу, которая будет запрашивать у пользователя год рождения и выводить ассоциированное с ним название животного по китайскому гороскопу. При этом программа не должна ограничиваться только

годами из приведенной таблицы, а должна корректно обрабатывать все годы нашей эры.

```
year = int (input('Введите год'))
if year < 0:
    print('такого года нет')
elif year % 12 == 0:
    print ('Обезьяна')
elif year % 12 == 1:
    print ('Петух')
elif year % 12 == 2:
    print ('собака')
elif year % 12 == 3:
    print ('свинья')
elif year % 12 == 4:
    print ('крыса')...
```

Домашнее задание.

1. Продолжить гороскоп (Задание 2).
2. Напишите программу, которая по номеру дня недели выводит его название.
3. Напишите программу, которая по номеру месяца в году выводит его название.