УНИВЕРСИТЕТ ИТМО

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

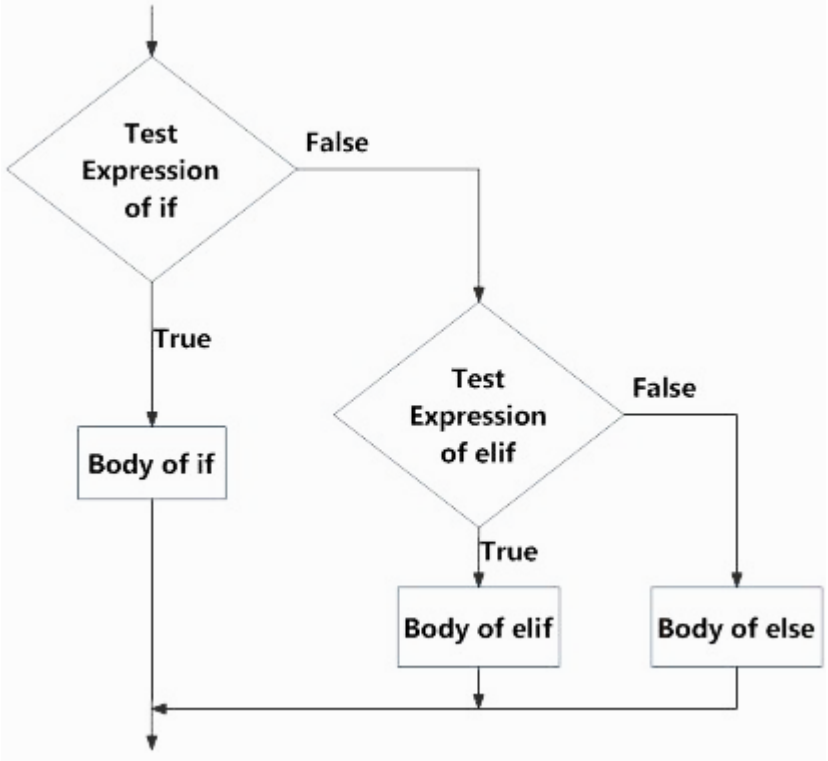
Python. Операторы ветвления и циклы.

Мусаев Андрей Александрович
amusayev1990@gmail.com

If-elif-else

Условная инструкция if-elif-else (её ещё иногда называют оператором ветвления) - основной инструмент выбора в Python.

```
if test1:
    state1
elif test2:
    state2
else:
    state3
```



If-elif-else

```
a = int(input())
if a < 18:
        print('Kid')
elif 18 <= a <= 21:
        print('Youth')
else:
        print('Adult')
```

If-elif-else

Проверка истинности в Python

Любое число, не равное 0, или непустой объект - истина.

Числа, равные 0, пустые объекты и значение None – ложь

Операции сравнения применяются к структурам данных рекурсивно

Операции сравнения возвращают True или False

Логические операторы and и or возвращают истинный или ложный объект-операнд

If-elif-else

Логические операторы:

X and Y
Истина, если оба значения X и Y истинны.

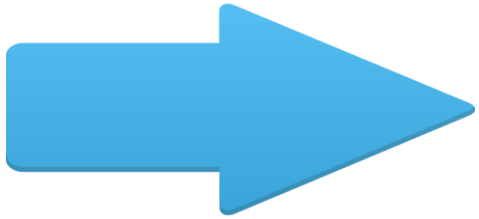
X or Y
Истина, если хотя бы одно из значений X или Y истинно.

not X
Истина, если X ложно.

If-elif-else

Трехместное выражение if/else

```
if X:
    A = Y
else:
    A = Z
```



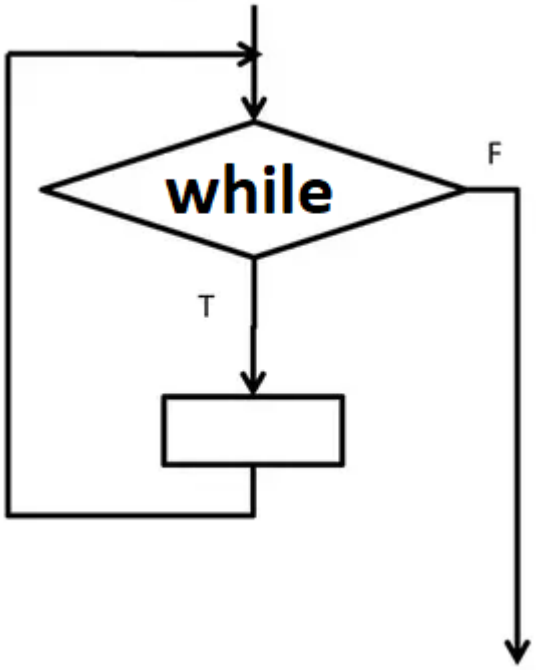
```
A = Y if X else Z
```

While

While - один из самых универсальных циклов в Python, поэтому довольно медленный. Выполняет тело цикла до тех пор, пока условие цикла истинно.

while test:

state(s)



While

```
i = 1
while i <= 10:
    print(i ** 2)
    i += 1
```

```
n = int(input())
length = 0
while n > 0:
    n //= 10
    length += 1
    print(length)
print(length)
```

For

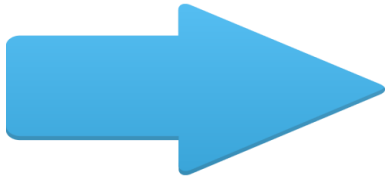
Для повторения цикла некоторое заданное число раз n можно использовать цикл for вместе с функцией range:

```
for i in range(4):                # равносильно инструкции for i in 0, 1, 2, 3
    print(i)
    print(i ** 2)
    print('Конец цикла')
```

Вызов range(a, b) означает, что индексная переменная будеть принимать значения от a до b - 1.

For

```
for i in 1, 2, 3, 'one', 'two', 'three':
    print(i)
```



```
1
2
3
'one‘
'two‘
'three'
```

```
i = 1
for color in 'red', 'orange', 'yellow', 'green', 'cyan', 'blue', 'violet':
    print('#', i, ' color of rainbow is ', color)
    i += 1
```

Continue

Оператор continue начинает следующий проход цикла, минуя оставшееся тело цикла (for или while)

```
for i in 'hello world':
    if i == 'o':
        continue
    print(i * 2, end='')
```

hheellll wwrrlldd

Break

Оператор break досрочно прерывает цикл.

```
for i in 'hello world':
    if i == 'o':
        break
    print(i * 2, end='')
```

hheellll

Else с циклом

Слово else, примененное в цикле for или while, проверяет, был ли произведен выход из цикла инструкцией break, или же "естественным" образом. Блок инструкций внутри else выполнится только в том случае, если выход из цикла произошел без помощи break.

```
for i in 'hello world':
    if i == 'a':
        break
else:
    print('Буквы a в строке нет')
```

Буквы a в строке нет