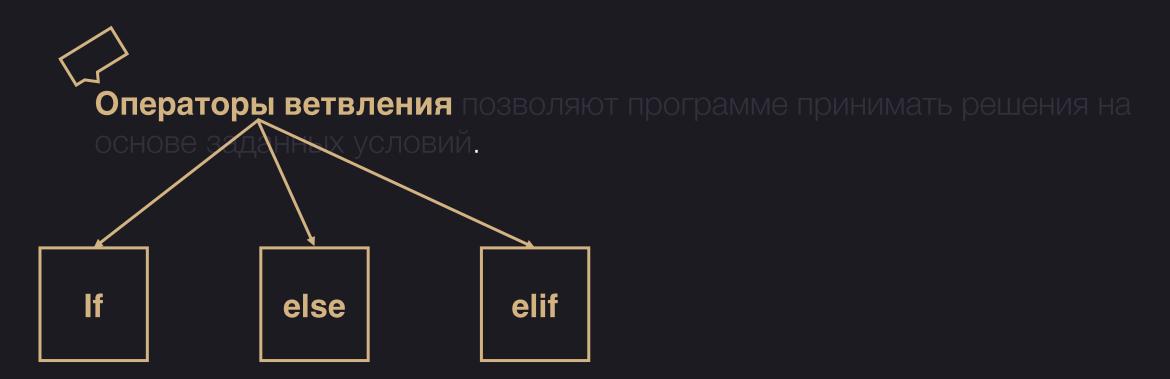


Операторы ветвления позволяют программе принимать решения на основе заданных условий.





Оператор if используется для выполнения кода, если определенное условие истинно.

```
••••

if условие:

# код, который выполнится, если условие истинно
```

If | Tpi

Пример.

```
возраст = 18
if возраст >= 18:
    print("Вы совершеннолетний")
```

else

Оператор else используется вместе с if для выполнения кода, когда условие ложно.

```
if условие:
    # код, если условие истинно
else:
    # код, если условие ложно
```

else

Пример.

```
Bospact = 16
if вospact >= 18:
    print("Вы совершеннолетний")
else:
    print("Вы несовершеннолетний")
```

elif

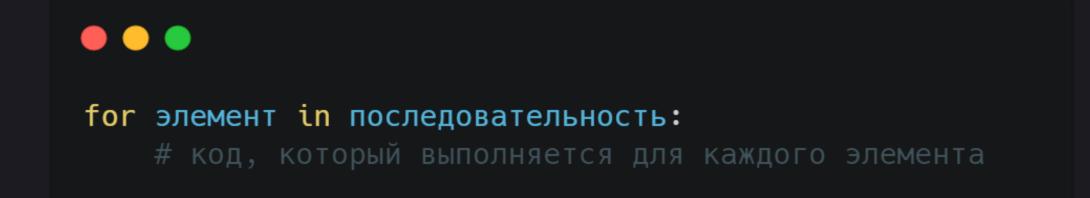
Оператор elif (сокращение от "else if") используется для проверки множества условий.

```
if условие1:
elif условие2:
elif условие3:
else:
    # код, если все условия ложны
```

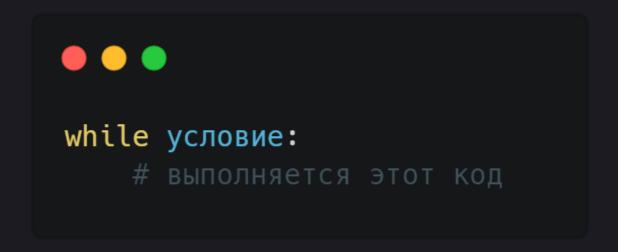
elif

Пример.

```
оценка = 4
if оценка == 5:
    print("Отлично")
elif оценка == 4:
    print("Xopowo")
elif оценка == 3:
    print("Удовлетворительно")
else:
    print("Неудовлетворительно")
```



```
cymma = 0
for число in range(1, 101):
    cymma += число
print(f"Cymma чисел от 1 до 100: {cymma}")
```





```
a = 5
while a > 0:
    print(a)
    a -= 1
```

break используется для немедленного выхода из цикла continue используется для перехода к следующей итерации цикла

```
for i in range(10):
   if i == 5:
      break
   print(i)
```

```
for i in range(10):
   if i % 2 == 0:
      continue
   print(i)
```

Условные операторы и циклы могут быть вложенными друг в друга для создания более сложной логики.

```
возраст = 25
имеет_лицензию = True
if возраст >= 18:
    if имеет_лицензию:
        print("Вы можете водить автомобиль")
    else:
        print("Вам нужно получить водительские права")
else:
    print("Вы слишком молоды для вождения")
```

Условные операторы и циклы могут быть вложенными друг в друга для создания более сложной логики.

```
for i in range(1, 6):
    for j in range(1, 6):
        print(f"{i} * {j} = {i*j}", end="\t")
    print()
```

