Самое важное

Оператор if
Этот оператор используется для составления условных блоков, которые нужны для управления потоком выполнения кода.

Синтаксис использования *if x > 0*: сначала идёт сам оператор if, затем какое-либо выражение, результат которого можно трактовать как правду или ложь.
Условное выражение может быть как большим и сложным, так маленьким и простым.
Примеры простых условий:

- if True: True/False — это результат любых условий, но никто не запрещает нам использовать их напрямую;
- if 1: — единица считается Python эквивалентом True;
- if 0: — ноль считается эквивалентом False.

Пример проверки выражения: число после каких-либо операций оказалось не равным 0.

x = 0
if x: — не сработает
…

x += 1
if x: — теперь сработает

Оператор else
Оператор else позволяет собрать все остальные возможные варианты.
Например, x равен случайному числу от 1 до 100. if x == 1: — так мы проверим, что x равен 1.
…
else: — так мы проверим, что x равен всем остальным числам, кроме 1.
....

При написании кода можно записывать операторы сравнения следующим образом.

Операции сравнения		Примеры использования
>	Больше	if a > b:
<	Меньше	if a < b:
>=	Больше либо равно	if a >= b:
<=	Меньше либо равно	if a <= b:
==	Равно	if a == b:
!=	Не равно	if a != b:

Не допускай следующих ошибок!

Оператор else не нуждается в условии.

```
if x == 1:
…
else x == 2:  — такой код будет ошибкой.
...

if x == 1:
…
else:
x == 2 — это условие никак не будет влиять на работу else или алгоритма в целом.
...
```

Не забывайте создавать переменную ДО условного блока.

```
start = 0
if start > 0: — старт равен 0, значит, условие не сработает.
   finish = 5 — переменная не будет создана.
```

print(finish) — тут вылезет ошибка, так как Python не знает о такой переменной, ведь код finish = 5 не был выполнен