

Питоша Python: работа на уроке

Фамилия, имя: _____

Дата: _____

Класс: _____

После тренажера: списки, цикл **for**, условие **if**, накопитель, максимум, **append**.

- 45 минут
- можно в парах
- без компьютера или с Python

Цель работы. Научиться не только собирать строки кода, но и объяснять, что происходит в программе: как меняются переменные, почему срабатывает условие и как получить новый список результатов.

1. Быстрая разминка: вспомни прием

Соедини команду и смысл. Впиши букву рядом с номером.

Команда	Ответ	Смысл
1) <code>for n in numbers:</code>		А. Добавить новый элемент в список
2) <code>total = total + n</code>		В. Пройти по всем элементам списка
3) <code>n % 2 == 0</code>		С. Увеличить накопитель на очередное число
4) <code>results.append("зачет")</code>		Д. Проверить, что число четное

Коротко объясни своими словами, зачем в программе нужна переменная-накопитель.

2. Предскажи вывод программы

А

```
fruits = ["яблоко", "банан", "груша"]
for fruit in fruits:
    print(fruit)
```

В

```
numbers = [4, 7, 2, 9]
total = 0
for n in numbers:
    total = total + n
print(total)
```

С

```
scores = [55, 82, 60, 41, 90]
count = 0
for s in scores:
    if s >= 60:
        count = count + 1
print(count)
```

Если работа идет за компьютером, сначала предскажи ответ на бумаге, потом проверь запуском.

Трассировка: программа как таблица

Трассировка помогает увидеть программу по шагам. После каждой итерации запиши, что стало с главной переменной.

3. Найди самое большое число

```
numbers = [3, 10, 6, 12]
biggest = numbers[0]
for n in numbers:
    if n > biggest:
        biggest = n
print(biggest)
```

Шаг	n	Условие n > biggest	biggest после шага	Комментарий
1	3			
2	10			
3	6			
4	12			

Что напечатает программа? _____

4. Построй список результатов

```
scores = [75, 58, 91, 60]
results = []
for s in scores:
    if s >= 60:
        results.append("зачет")
    else:
        results.append("нет")
print(results)
```

s	Условие s >= 60	Что добавится в results	results после шага
75			
58			
91			
60			

Итоговый вывод программы:

Ошибки, которые часто прячутся в Python

5. Исправь программы

В каждой программе есть ошибка. Найди ее, подпиши причину и запиши исправленный вариант.

А. Потерян один знак

```
numbers = [2, 5, 8]
for n in numbers
    print(n)
```

Причина ошибки:

В. Съехал блок

```
scores = [40, 70, 90]
count = 0
for s in scores:
    if s >= 60:
        count = count + 1
print(count)
```

Причина ошибки:

С. Список заменяется строкой

```
results = []
scores = [50, 80]
for s in scores:
    if s >= 60:
        results = "зачет"
    else:
        results = "нет"
print(results)
```

Как сделать так, чтобы в списке осталось два результата?

6. Мини-проект: журнал зачета

Дан список баллов:

```
scores = [72, 45, 88, 60, 39, 91]
```

Напиши программу, которая:

☐ создает список **results** со словами "зачет" или "нет";

☐ находит лучший балл;

☐ считает, сколько человек получили зачет;

☐ печатает **results**, количество зачетов и лучший балл.

Расширение: сделай программу умнее

7. Три уровня результата

Доработай мини-проект. Теперь результат должен быть таким:

Балл	Что добавить в results
85 и выше	"отлично"
от 60 до 84	"зачет"
меньше 60	"нет"

Подсказка: внутри цикла можно использовать `if / elif / else`.

8. Проверка понимания

Ответь коротко.

1) Почему `count = 0` пишут до цикла, а не внутри?

2) Что изменится, если в условии заменить `s >= 60` на `s > 60`?

3) Где в твоей программе можно ошибиться с отступами?

Самооценка

База

☐ Я могу прочитать цикл и предсказать вывод.

Уверенно

☐ Я могу заполнить таблицу трассировки и объяснить изменение переменной.

Сильный шаг

☐ Я могу написать программу со списком, условием и накопителем.