



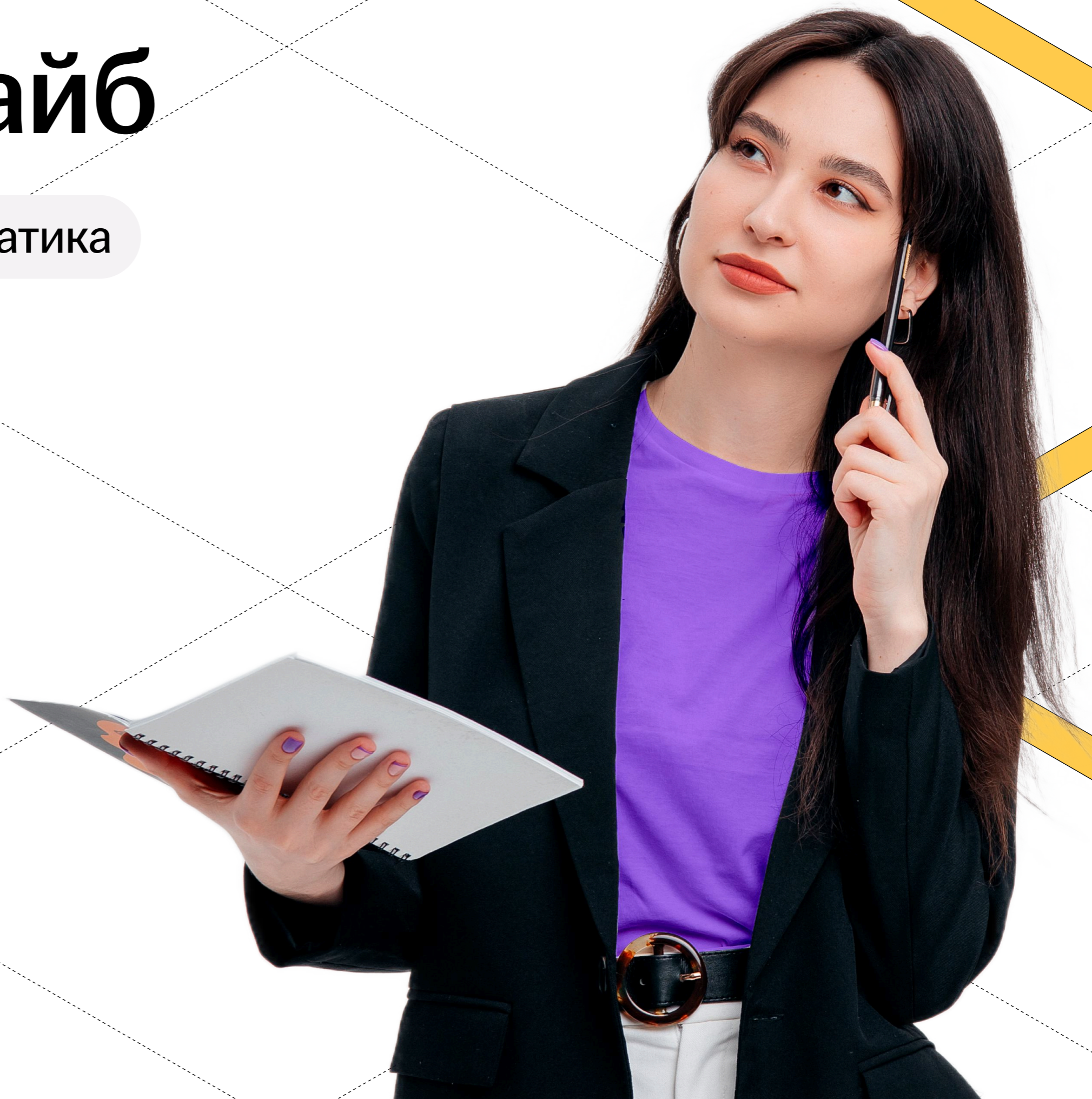
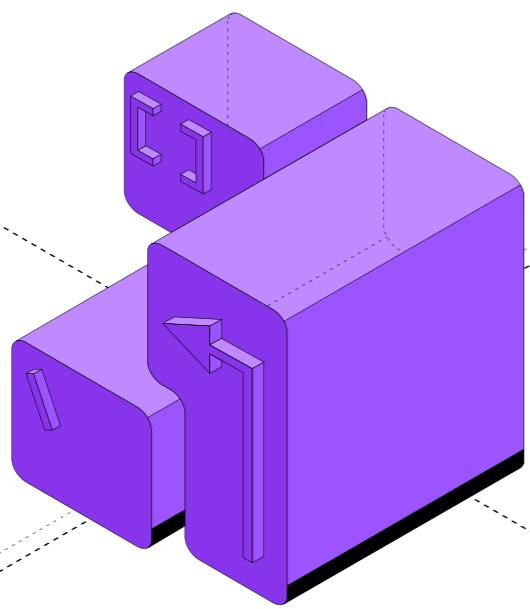
Конспект занятия

База Python

Софья Вайб

10 класс

Информатика





Скачивание и установка Python

1. Перейдите на [страницу загрузки Python](#).
2. Выберите последнюю версию Python для вашей операционной системы (Windows, macOS, или Linux) и нажмите на соответствующую кнопку загрузки.

Looking for a specific release?
Python releases by version number:

Release version	Release date	Click for more	
Python 3.12.5	Aug. 6, 2024	Download	Release Notes
Python 3.12.4	June 6, 2024	Download	Release Notes
Python 3.12.3	April 9, 2024	Download	Release Notes
Python 3.11.9	April 2, 2024	Download	Release Notes
Python 3.10.14	March 19, 2024	Download	Release Notes
Python 3.9.19	March 19, 2024	Download	Release Notes
Python 3.8.19	March 19, 2024	Download	Release Notes
Python 3.11.8	Feb 6, 2024	Download	Release Notes

[View older releases](#)

3. Запустите скачанный установочный файл.
4. В установщике Python обязательно отметьте галочку "Add Python to PATH", чтобы Python автоматически добавился в системные переменные окружения.
5. Нажмите "Install Now" для начала установки.
6. Дождитесь завершения установки и закройте установщик.

Скачивание и установка PyCharm

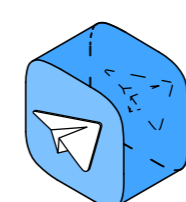
1. Перейдите на [страницу загрузки PyCharm](#).
2. Выберите версию PyCharm для вашей операционной системы (Windows, macOS, или Linux) и нажмите на кнопку загрузки.

Другие версии

Версия 2024.2

PyCharm Professional Edition	PyCharm Community Edition
2024.2.1 - Linux (tar.gz)	2024.2.1 - Linux (tar.gz)
2024.2.1 - Linux ARM64 (tar.gz)	2024.2.1 - Linux ARM64 (tar.gz)
2024.2.1 - Windows (exe)	2024.2.1 - Windows (exe) ←
2024.2.1 - Windows ARM64 (exe)	2024.2.1 - Windows ARM64 (exe)
2024.2.1 - ZIP archive (win.zip)	2024.2.1 - ZIP archive (win.zip)
2024.2.1 - ZIP archive for Windows ARM64 (win.zip)	2024.2.1 - ZIP archive for Windows ARM64 (win.zip)
2024.2.1 - macOS (dmg)	2024.2.1 - macOS (dmg)
2024.2.1 - macOS Apple Silicon (dmg)	2024.2.1 - macOS Apple Silicon (dmg)

3. Запустите скачанный установочный файл.
4. Следуйте инструкциям установщика для завершения установки PyCharm.





Типы данных и переменные

Типы данных

→ Числа:

- int — целые числа
- float — вещественные числа

→ Строки:

- str

→ Списки:

- list

→ Логический тип:

- bool

Переменная — контейнер для данных.

<имя_переменной> = <значение>



Таким образом, значение, указанное в переменной, можно использовать, обратившись к нему по имени переменной.

Пример:

Основные операторы

=	оператор присваивания
+	сложение
-	вычитание
*	умножение
**	возведение в степень
/	деление
%	остаток от деления
//	целая часть от деления

Сокращенная форма записи математических операторов

a += x	↔	a = a + x
a *= x	↔	a = a * x
a %= x	↔	a = a % x
a += x + y	↔	a = a + (x + y)
a *= x + y	↔	a = a * (x + y)
a %= x + y	↔	a = a % (x + y)

Полезные встроенные функции

abs() — модуль числа.

min() — поиск минимального элемента.

int() — приведение к целому десятичному числу.

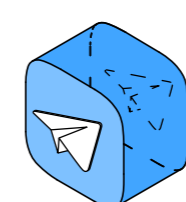
sum() — подсчет суммы (для списка!).

max() — поиск максимального элемента.

Ввод, вывод и пример программы

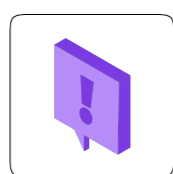
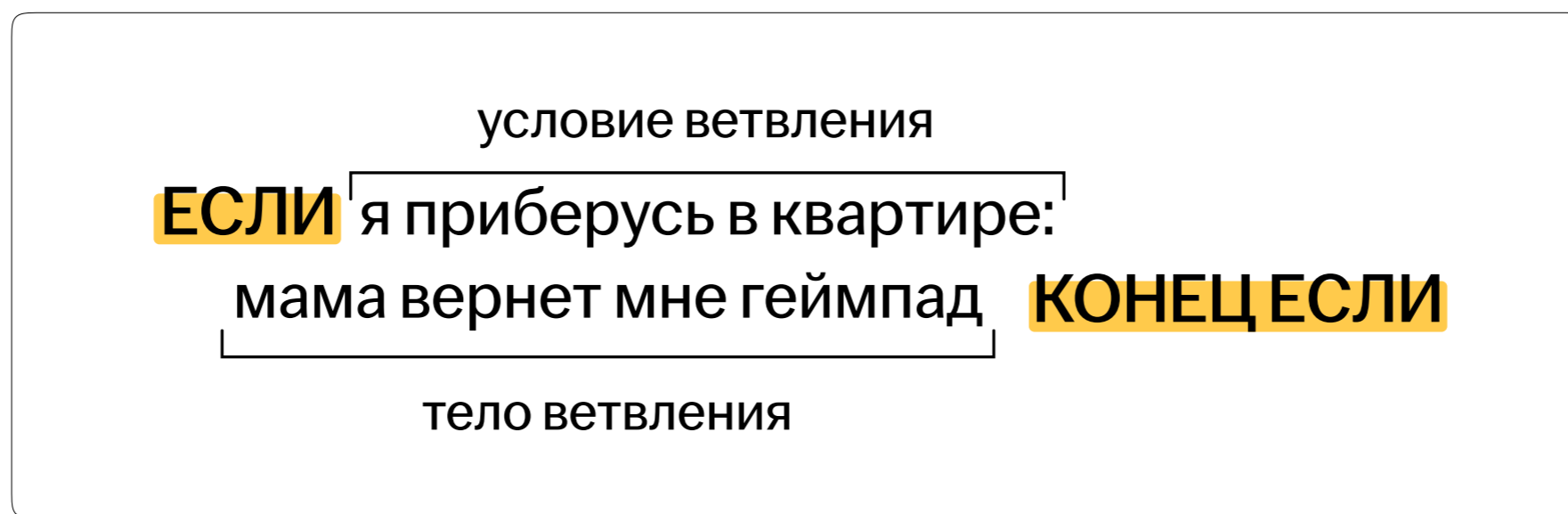
Вывод данных на экран производится с помощью команды **print()**.

Ввод данных с клавиатуры производится с помощью команды **input()**.





Ветвление



Команды тела ветвления выполняются только тогда, когда выполняется его условие.

Запись ветвления в Python

```
if <условие>:
    <набор команд 1>
else:
    <набор команд 2>
```

Пример:

Каскадное ветвление if-elif-else

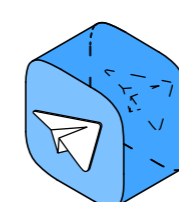
```
if <условие 1>:
    <набор команд 1>      условие 1 ✓
elif <условие 2>:
    <набор команд 2>      условие 1 ✗
                        условие 2 ✓
...
elif <условие n>:
    <набор команд n>      все условия выше ✗
                        условие n ✓
else:
    <дефолтный набор команд>
                        все условия ✗
```

Используется для обработки взаимоисключающих условий.

Пример: определение знака числа.



При этом бывает так, что для решения одной задачи можно воспользоваться как каскадным, так и вложенным ветвлением, они не взаимоисключают друг друга.

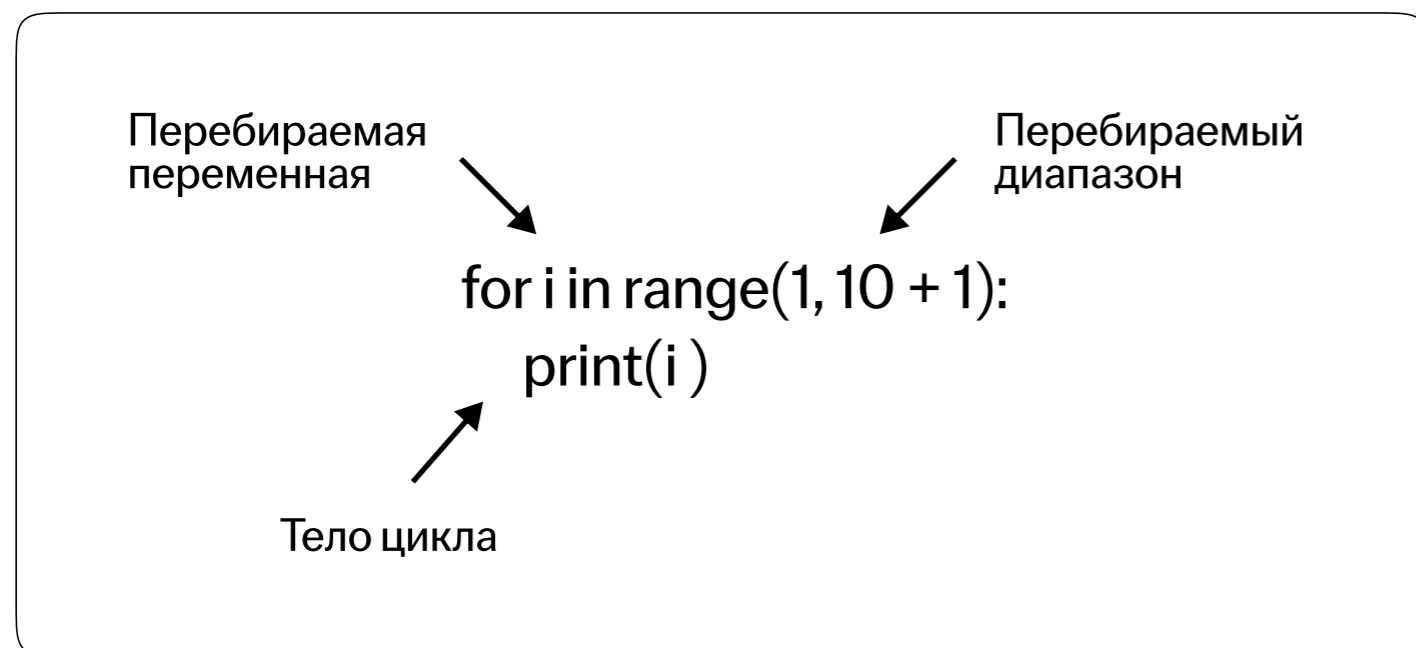




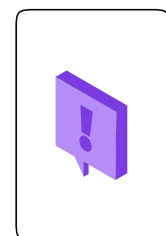
Определение цикла

Цикл — многократное выполнение определенного набора команд.

Цикл for



- На каждом новом шаге цикла перебираемая переменная будет принимать новое значение из перебираемого диапазона. В данном примере на первом шаге цикла $i = 1$. на втором $i = 2$ и так далее.
- Команды тела цикла выполняются с каждым новым i на каждом шаге цикла, пока перебор не дойдет до последнего элемента.



Важный момент:

Диапазон `range(a, b)` перебирает целые числа из диапазона от a до $b - 1$ включительно. Чтобы перебрать в том числе самое последнее значение b , диапазон стоит увеличить на 1: `range(a, b + 1)`.

Цикл for

- цикл перебора;
- удобен, когда известно точное количество повторений или есть конкретный перебираемый набор.

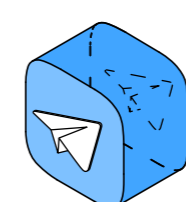
Расширенный функционал range

- `range(n)` — перебор чисел от 0 до $n - 1$;
- `range(a, b, n)` — перебор чисел от a до $b - 1$ с шагом n . Например, `range(2, 10, 3)` переберет числа 2, 5, 8;
- в том числе с отрицательным n можно запустить перебор от большего к меньшему. Например, `range(10, 5, -1)` переберет числа 10, 9, 8, 7, 6.

break и continue

break и **continue** — операторы прерывания цикла.

- Если в цикле встречается команда `continue`, цикл пропускает все последующие действия данного шага и переходит к следующему.
- Если в цикле встречается команда `break`, он моментально завершает всю свою работу.





break и continue

for что-нибудь in где-нибудь:

```
...
...
continue
```

```
...
...
```

```
...
...
```

for что-нибудь in где-нибудь:

```
...
...
break
```

```
...
...
```

```
...
...
```

```
for i in range(1, 10):
    if i == 5:
        continue
    print(i)
```

Вывод: 1, 2, 3, 4, 6, 7, 8, 9.

```
for i in range(1, 10):
    if i == 5:
        break
    print(i)
```

Вывод: 1, 2, 3, 4.

Определение цикла

Цикл — многократное выполнение определенного набора команд.

Цикл while

```
a = 10
while a >= 1:
    print(a)
    a -= 1
```

Условие работы цикла

Тело цикла

→ Команды тела цикла будут последовательно выполняться, пока условие цикла истинно.

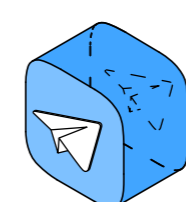
Разница между циклами

Цикл for

- цикл перебора;
- удобен, когда известно точное количество повторений или есть конкретный перебираемый набор.

Цикл while

- цикл по условию;
- удобен, когда точное количество повторений не известно, но известно, до какого момента необходимо повторять действия.



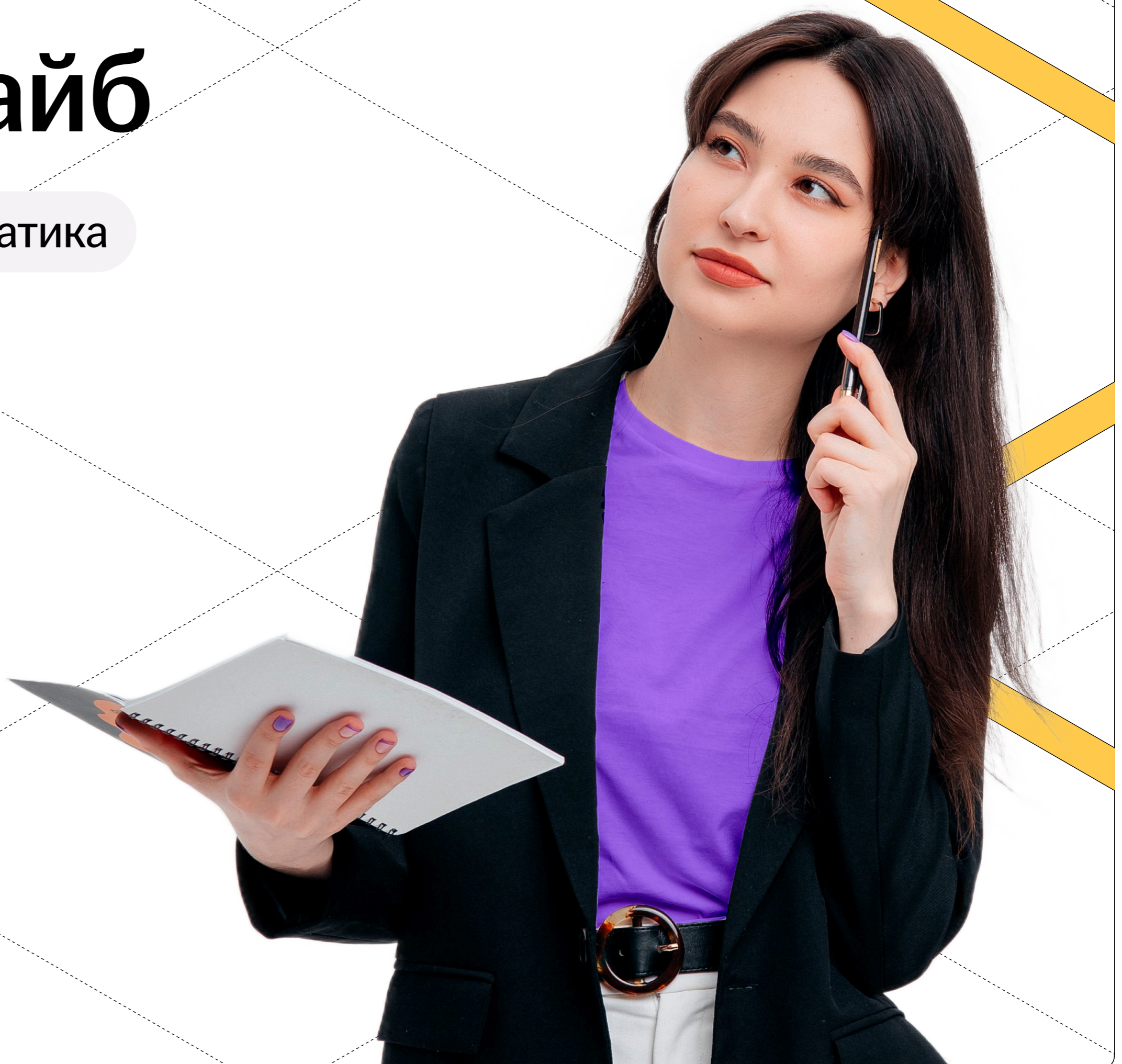
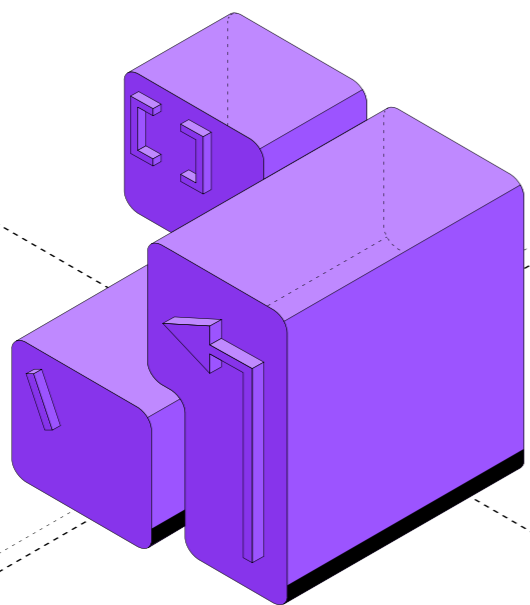
Конспект занятия

Электронные таблицы

Софья Вайб

10 класс

Информатика





Ввод значений и формул в таблице Excel

Значения просто вписываются в ячейку

	A
1	55+6

Формулы начинаются со знака =

	A
1	=55+6

→

	A
1	61

Адресация в Excel

Значения в ячейках можно использовать по адресу ячеек:

	A	B
1	5	10
2		
3		=A1+B1

→

	A	B
1	5	10
2		
3		15

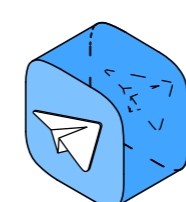
Относительная адресация

	A	B	C	D
1	5	10	=A1+B1	
2	10	15		
3	15	20		
4	20	25		
5	25	30	1	15
6	30	35	2	
7	35	40	3	
8	40	45	4	

	A	B	C	D
1	5	10		15
2	10	15		25
3	15	20		35
4	20	25		45
5	25	30		55
6	30	35		65
7	35	40		75
8	40	45		85

При копировании формулы с относительной адресацией ссылки на ячейки подстраиваются.

Так, в ячейке D1 были ссылки на A1 и B1, при копировании в ячейку ниже ссылки изменились на A2 и B2, еще на одну ниже — на A3 и B3 и т. д.





Базовые функции

СУММ(диапазон) — находит сумму значений из указанного диапазона;

СЧЁТ(диапазон) — находит количество значений в указанном диапазоне;

МИН(диапазон) — находит наименьшее значение из диапазона;

МАКС(диапазон) — находит наибольшее значение из диапазона;

СРЗНАЧ(диапазон) — находит среднее арифметическое значений диапазона.

Основные математические функции

СУММ(диапазон) — находит сумму значений из указанного диапазона;

СЧЁТ(диапазон) — находит количество значений в указанном диапазоне;

ПРОИЗВЕД(диапазон) — находит произведение значений из указанного диапазона;

МИН(диапазон) — находит наименьшее значение из диапазона;

МАКС(диапазон) — находит наибольшее значение из диапазона;

НАИБОЛЬШИЙ(диапазон; n) — находит n-ное наибольшее значение из диапазона;

НАИМЕНЬШИЙ(диапазон; n) — находит n-ное наименьшее значение из диапазона;

ОСТАТ(m, n) — находит остаток от деления m на n;

ABS(число) — находит модуль числа;

СРЗНАЧ(диапазон) — находит среднее арифметическое значений диапазона.

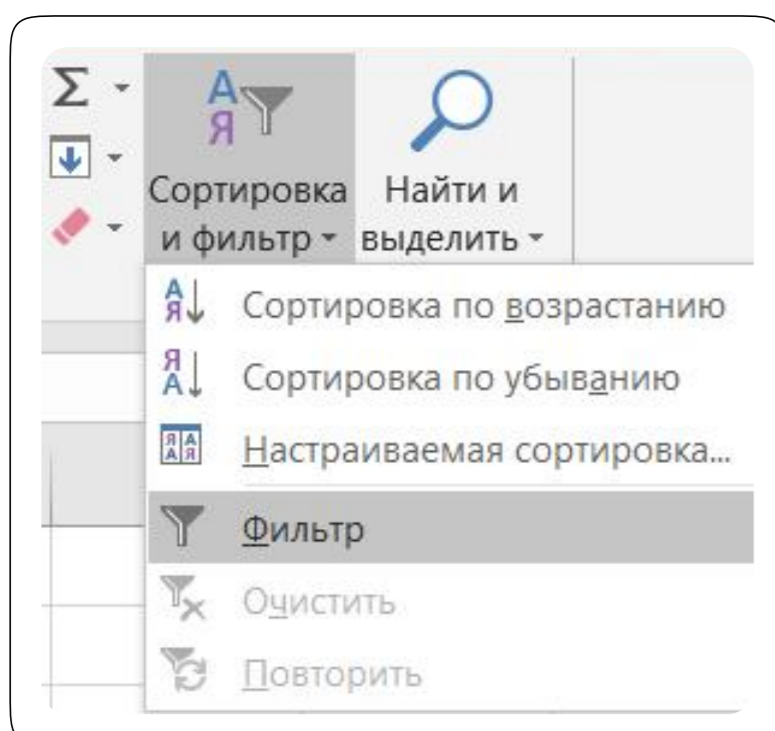
ОСНОВАНИЕ(число; основание) — переводит число из десятичной в другую систему счисления.

Включение фильтров

	A	B
1	Число 1	Число 2
2	35	16
3	2	68
4	71	12
5	90	8

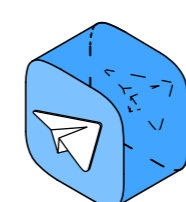
→ Перед включением фильтрации необходимо выбрать диапазоны данных, которые должны быть отфильтрованы. Делается это простым выделением курсором.

→ Можно выделять целые столбцы, чтобы под фильтр попали все данные, находящиеся в них. Для корректной работы с данными каждому столбцу стоит иметь свой заголовок.



→ Включить фильтрацию можно в главном разделе верхнего меню, нажав на «Сортировка и фильтр» и выбрав «Фильтр».

→ После этого под заголовками появятся маленькие кнопки, по нажатию на которые откроется меню фильтрации.

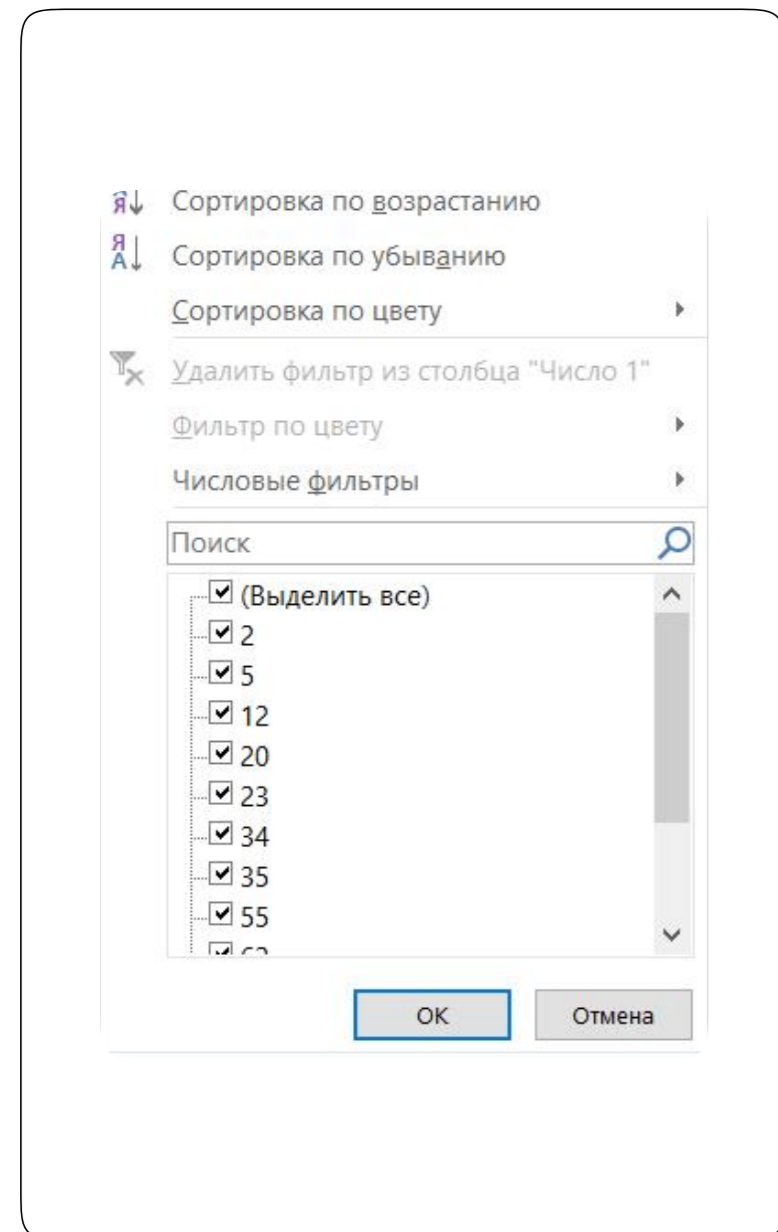




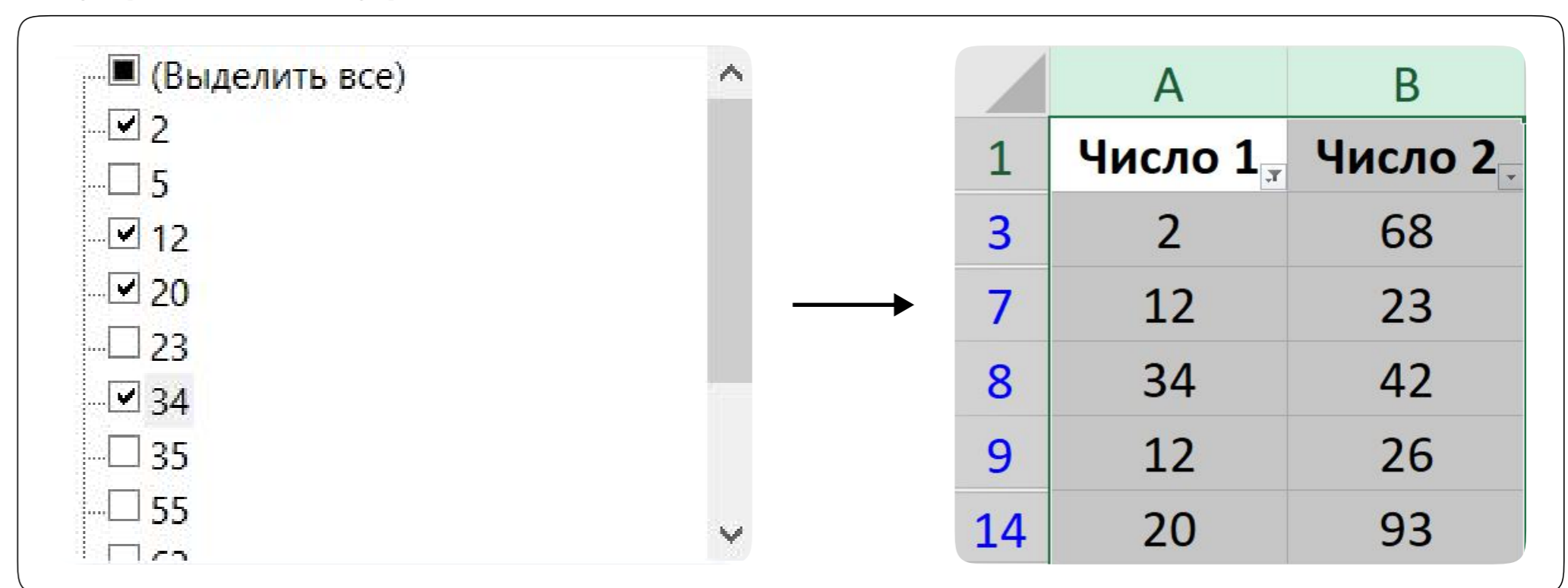
	A	B
1	Число 1	Число 2
2	35	16

- Если было выбрано несколько столбцов с данными, фильтрация будет происходить по тому столбцу, у заголовка которого была нажата кнопка фильтрации.
- При этом соседние числа останутся рядом и не перемешаются.

Меню фильтрации



- При нажатии на кнопку фильтрации вылезет следующее меню: “Сортировка по возрастанию” и “Сортировка по убыванию” расположат все данные в соответствующем порядке.
- Ниже в отдельном окне можно вручную выбрать, какие из данных должны быть показаны. Чтобы скрыть ненужные данные, достаточно убрать галочку рядом с ними.



Логические функции

ЕСЛИ(условие; действие 1; действие 2) — если условие выполняется, производится действие 1, иначе производится действие 2.

Следующие функции производят соответствующие логические операции над переданными условиями.

И(условие 1; условие 2; ...);

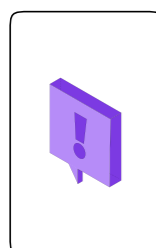
ИСКЛЮЧИЛИ(условие 1; условие 2; ...);

ИЛИ(условие 1; условие 2; ...);

НЕ(условие).

СУММЕСЛИ(диапазон; условие) — находит сумму элементов из указанного диапазона, которые подходят под условие.

СЧЁТЕСЛИ(диапазон; условие) — находит количество элементов из указанного диапазона, которые подходят под условие.



Важно!

не содержащие ссылки элементы условия должны быть в кавычках, ссылки приписываются к ним после символа &.

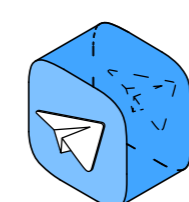
Например:

СЧЁТЕСЛИ(диапазон; “5”) — считает количество элементов, равных 5;

СЧЁТЕСЛИ(диапазон; “>5”) — считает количество элементов, больше 5;

СЧЁТЕСЛИ(диапазон; A1) — считает количество элементов, равных значению в ячейке A1;

СЧЁТЕСЛИ(диапазон; “>”&A1) — считает количество элементов, которые больше значения в ячейке A1.



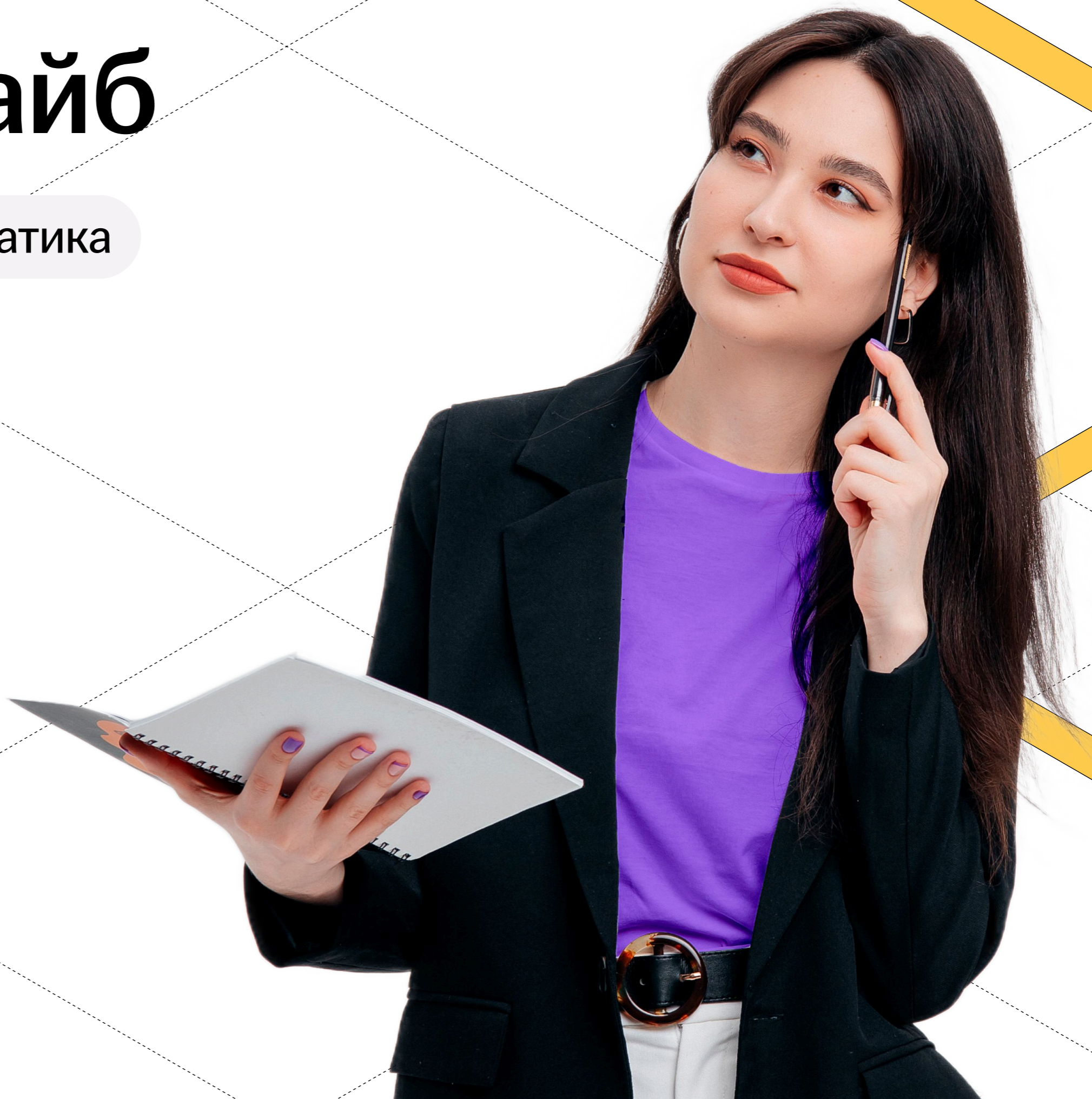
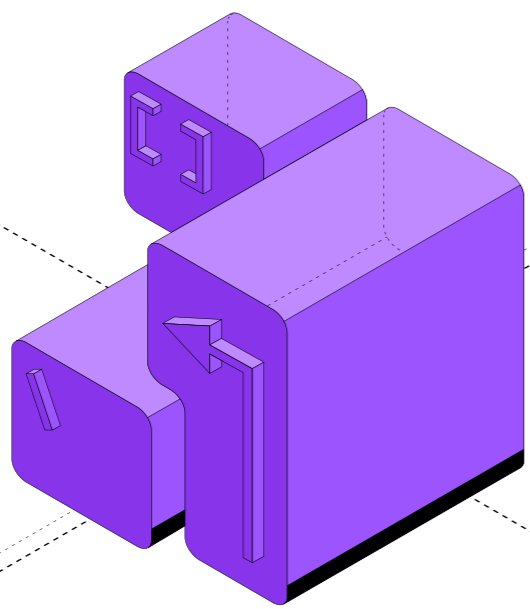
Конспект занятия

Алфавитный подход. Посимвольное кодирование

Софья Вайб

10 класс

Информатика





Определения

Язык — это система знаков для передачи, хранения и обработки информации.

Алфавит — это набор знаков в языке.

Мощность алфавита — количество знаков в алфавите.

Сообщение — последовательность знаков алфавита.

Бит — наименьшая единица измерения информации, которую можно записать в виде цифры 0 или 1.

Алфавитный подход



Алфавитный подход к измерению информации рассматривает информацию не с точки зрения содержания, а как последовательность символов.

$$N = 2^i$$

i — количество информации, которое несет 1 символ/элемент.

N — мощность алфавита.

По этой формуле все символы алфавита кодируются одинаковым и минимально возможным числом бит. В случае, если N не является степенью двойки, используется формула $N \leq 2^i$.

Важно!

При алфавитном подходе смысл сообщения не учитывается, количество информации определяется только длиной сообщения и мощностью алфавита.

Информационный объем текста

$$I = K * i$$

i — информационный объем одного символа.

I — информационный объем текста.

K — количество символов в тексте.

