СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Основы языка Python. Порядок выполнения программы

Преподаватель: старший преподаватель кафедры «Прикладная информатика»
Фельдман А.Г.

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Области применения

Системное программирование

Windows Mac OSX Linux

Игры, искусственный интеллект

Графический интерфейс

Qt

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

Программирование научных вычислений

python™

Веб-сценарии

django

Flask
web development, one drop at a time

Быстрое создание прототипов

Приложения баз данных

SQLite

Интеграция компонентов:
Cpython
Jython
IronPython

Java

Microsoft .NET

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

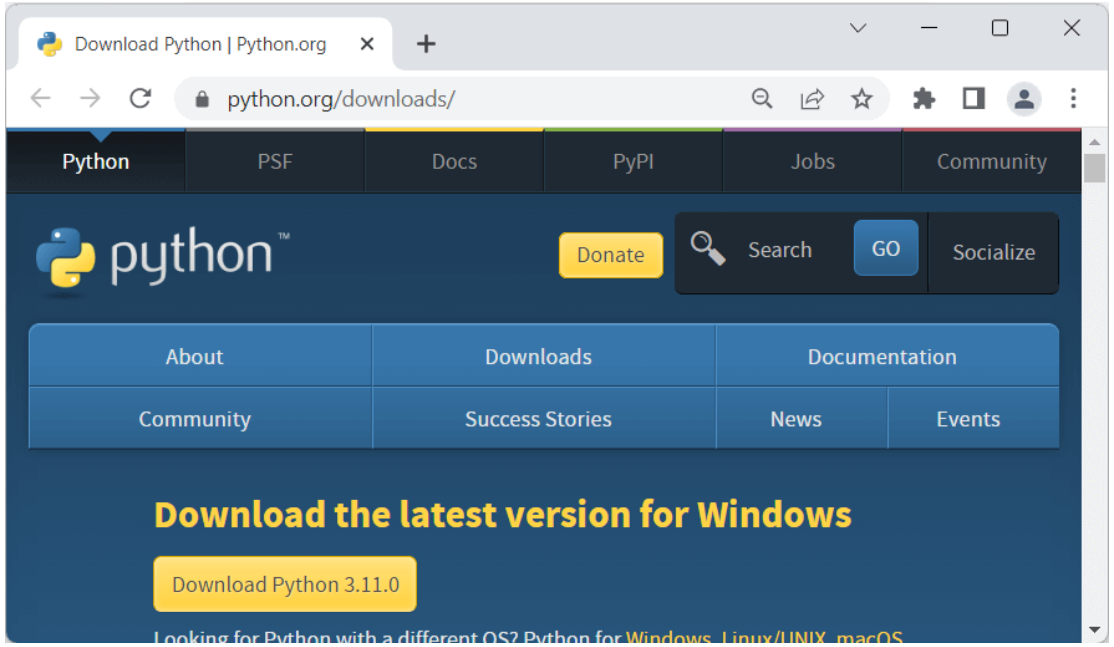
Особенности Python

- интерпретируемый язык
- простой в использовании язык
- язык с динамической типизацией
- язык высокого уровня
- объектно-ориентированный язык

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Установка Python и сред разработки

Для создания программ на **Python** нам потребуется интерпретатор, который доступен для установки на официальном сайте: https://www.python.org/downloads/



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Среды разработки

Для удобства разработки кода программисты используют среды разработки (IDE).

- Python IDLE;
- Sublime Text;
- Visual Studio Code;
- Jupyter Notebook;
- PyCharm.

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Первая программа

first program.py

```
1  # Эта программа приветствует пользователя и запрашивает ввод информации.
2
3  print('Здравствуй, мир!')
4  myName = input('Как тебя зовут? ') # запрос имени
5  print('Рад познакомиться с тобой, ' + myName)
6  print('Длина твоего имени:')
7  print(len(myName), 'букв(ы)')
8  myAge = input('Сколько тебе лет? ') # запрос возраста
9  print('Через год тебе будет ' + str(int(myAge) + 1) + ' год.')
```

```
Здравствуй, мир!
Как тебя зовут? Андрей
Рад познакомиться с тобой, Андрей
Длина твоего имени:
6 букв(ы)
Сколько тебе лет? 20
Через год тебе будет 21 год.
```

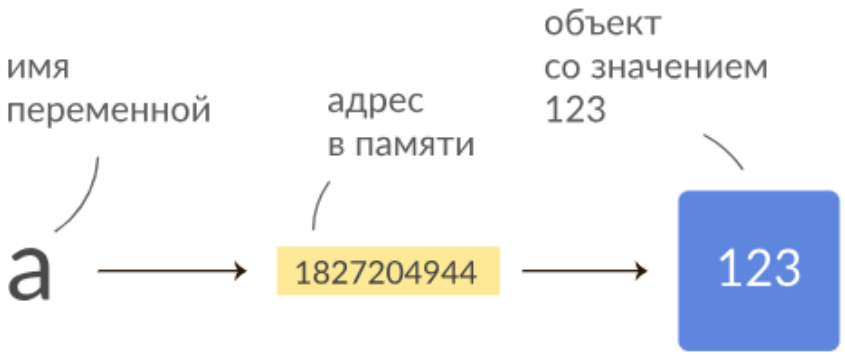
СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Переменные

Переменная – это имя, которое ссылается (указывает) на некоторое значение.

```
>>> n=17
>>> type(n)
<class 'int'>
>>> pi = 3.14159
>>> type(pi)
<class 'float'>
```

Тип переменной – это тип значения, на которое она указывает



СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Переменные

```
>>> more$=8
SyntaxError: invalid syntax
>>> 76fruits='apple'
SyntaxError: invalid decimal literal
>>> class='programming'
SyntaxError: invalid syntax
```

and	del	from	None	True
as	elif	global	nonlocal	try
assert	else	if	not	while
break	except	import	or	with
class	False	in	pass	yield
continue	finally	is	raise	async
def	for	lambda	return	await

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Типы данных

Тип данных	Примеры
Целые числа (int)	-2, -1, 0, 1, 2
Вещественные числа (float)	-1.25, -1.0, -0.5, 0.0, 0.5
Строки (str)	‘a’, ‘b’, ‘c’, ‘Hello’, ‘11’
Логические значения (bool)	True, False

```
>>> type(1)
<class 'int'>
>>> type(1.0)
<class 'float'>
>>> type('1.0')
<class 'str'>
```

```
>>> type(True)
<class 'bool'>
>>> type('True')
<class 'str'>
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Инструкции

Инструкция – это элемент кода, который интерпретатор Python может выполнить.

Например, скрипт

```
1  print(1)
2  x = 2
3  print(x)
```

выводит

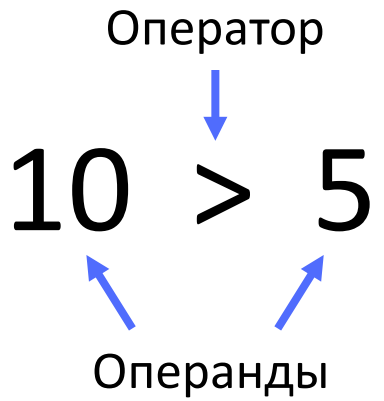
```
1
2

Process finished with exit code 0
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Операторы и операнды

Операторы – это специальные символы, представляющие вычислительные операции, например сложение и умножение. Значения, к которым применяется оператор, называются **операндами**.



СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Арифметические операторы

- `+` — сложение
- `-` — вычитание
- `*` — умножение
- `**` — возведение в степень
- `/` — деление
- `//` — целочисленное деление
- `%` — остаток от деления

Приоритеты операций:

Операции	Направление
**	Справа налево
* / // %	Слева направо
+ -	Слева направо

4 3 2 1 5

```
>>> number = 1 + 3 * 2 ** 3 ** 2  + 7
>>> print(number)
1544
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Операторы сравнения

Оператор	Операция
==	равно
!=	не равно
<	меньше
>	больше
<=	меньше или равно
>=	больше или равно

```
>>> 42==99
False
>>> 2!=2
False
>>> 'привет'=='Привет'
False
>>> 42==42.0
True
>>> 42=='42'
False
>>> 42>100
False
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Булевы операторы

Таблица истинности для оператора and

Выражение	Результат
True and True	True
True and False	False
False and True	False
False and False	False

```
>>> (4<5) and (5<6)
True
>>> (4<5) and (9<6)
False
```

Таблица истинности для оператора or

Выражение	Результат
True or True	True
True or False	True
False or True	True
False or False	False

```
>>> (1==2) or (2==2)
True
```

Таблица истинности для оператора not

Выражение	Результат
not True	False
not False	True

```
>>> not (5>3)
False
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Базовые алгоритмические структуры

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Условия

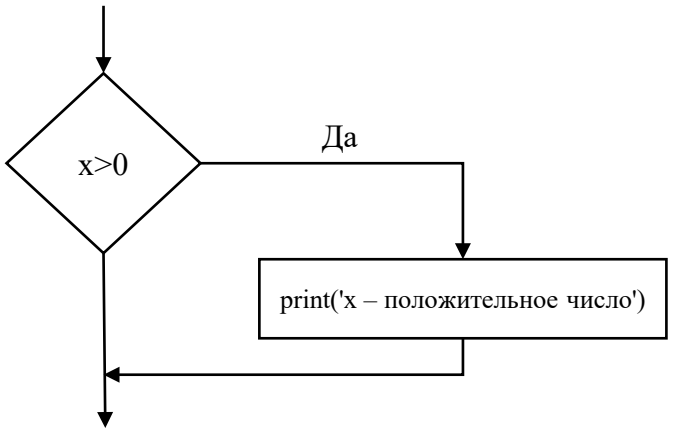
Синтаксис условного оператора:

```
if условие:
        действие 1
else:
        действие 2
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Условия

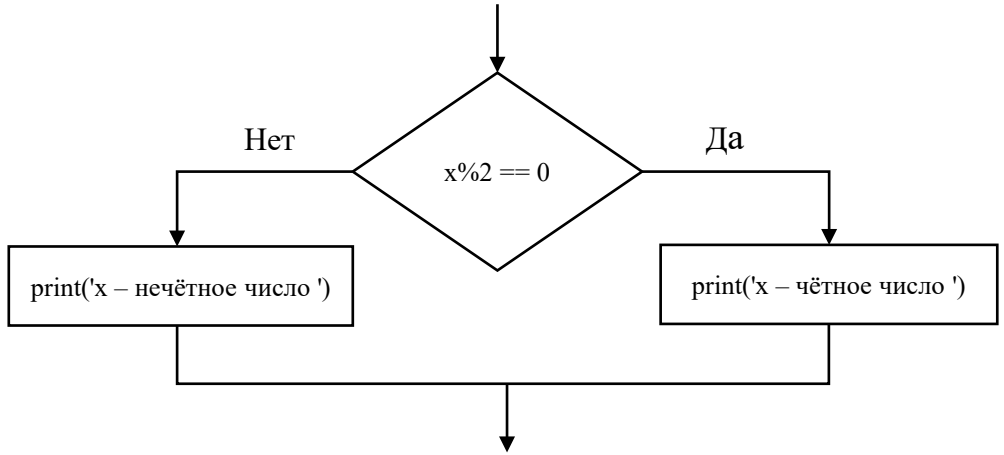
Сокращенная форма




```
1  x=int(input('Введите число: '))
2  if x>0:
3      print(f'{x} - положительное число')
```

```
Введите число: 15
15 - положительное число
```

Полная форма




```
1  x=int(input('Введите число: '))
2  if x%2==0:
3      print(f'{x} - чётное число')
4  else:
5      print(f'{x} - нечётное число')
```

```
Введите число: 15
15 - нечётное число
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

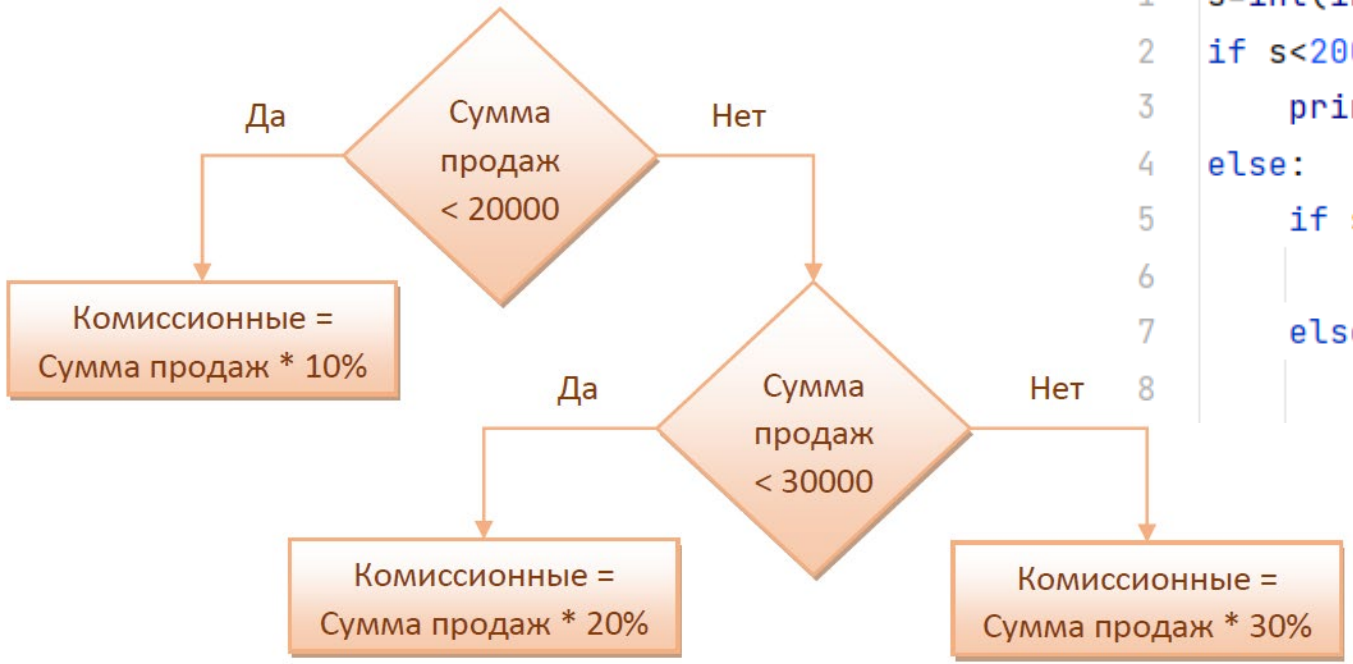
Вложенные условия

```
if условие 1:
        действие 1
else:
        if условие 2:
                действие 3
        else:
                действие 4
```

Вложенное условие

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Пример. Рассчитаем размер комиссионного вознаграждения продавцу: если сумма продаж меньше 20000 руб., то комиссионные составляют 10 % от этой суммы, если больше или равна 20000, но меньше 30000, то 20 %, а если больше 30000, то 30 %.




```
1  s=int(input('Введите сумму продаж продавца: '))
2  if s<20000:
3      print(f'Комиссионное вознаграждение {s*0.1} руб.')
4  else:
5      if s<30000:
6          print(f'Комиссионное вознаграждение {s * 0.2} руб.')
7      else:
8          print(f'Комиссионное вознаграждение {s * 0.3} руб.')
```

```
Введите сумму продаж продавца: 50000
Комиссионное вознаграждение 15000.0 руб.
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Каскадное ветвление

Если после else сразу следует еще один оператор if, можно использовать «каскадное» ветвление с ключевыми словами elif (сокращение от else if).

```
1  cost=1500
2  if cost<1000:
3      print('Скидок нет.')
4  elif cost<2000:
5      print('Скидка 2%.')
6  elif cost<5000:
7      print('Скидка 5%.')
8  else:
9      print('Скидка 10%.')
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Сложные условия

Логические операторы часто предоставляют возможность упрощения вложенных условных инструкций.

```
1    a=10
2    b=15
3    x=int(input('Введите значение x: '))
4    if x>=a and x<=b:
5        print('Принадлежит')
6    else:
7        print('Не принадлежит')
```

```
Введите значение x: 12
Принадлежит
```

Порядок вычислений:

1. отношения (<, <=, >, >=, ==, !=),
2. операции not,
3. операции and,
4. операции or.

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Сложные условия

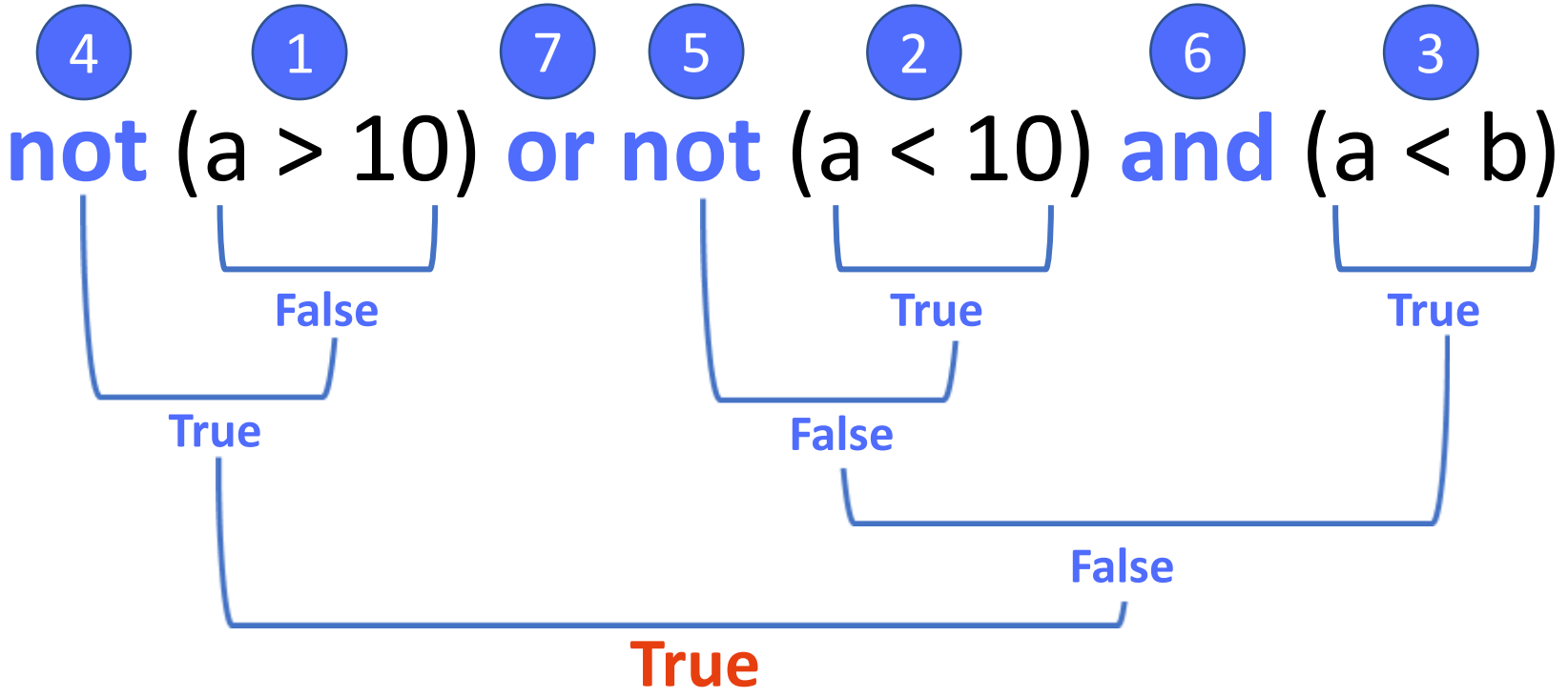
Задание 1. Запишите равносильные условия, не используя операцию НЕ:

a. not (a < 6) ⇔ a >= 6
b. not (b == c + d) ⇔ b != c + d
c. not (c != 15) ⇔ c == 15
d. not (7 < a **and** a < 12) ⇔ 7 >= a **or** a >= 12
e. not (b != c **or** d < 5) ⇔ b == c **and** d >= 5

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Сложные условия

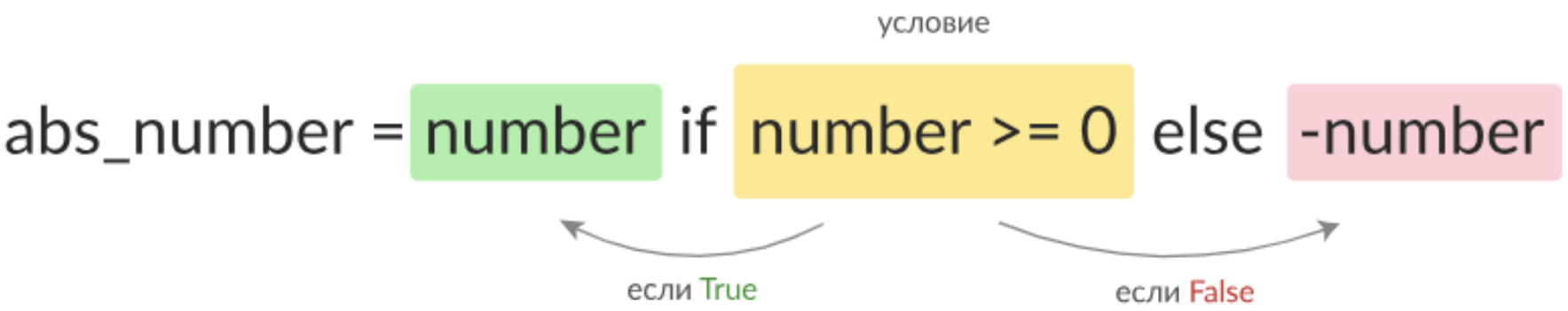
Задание 2. Определите, истинно или ложно следующее логическое выражение при **a = 5**, **b = 15**.



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

if else в одну строку

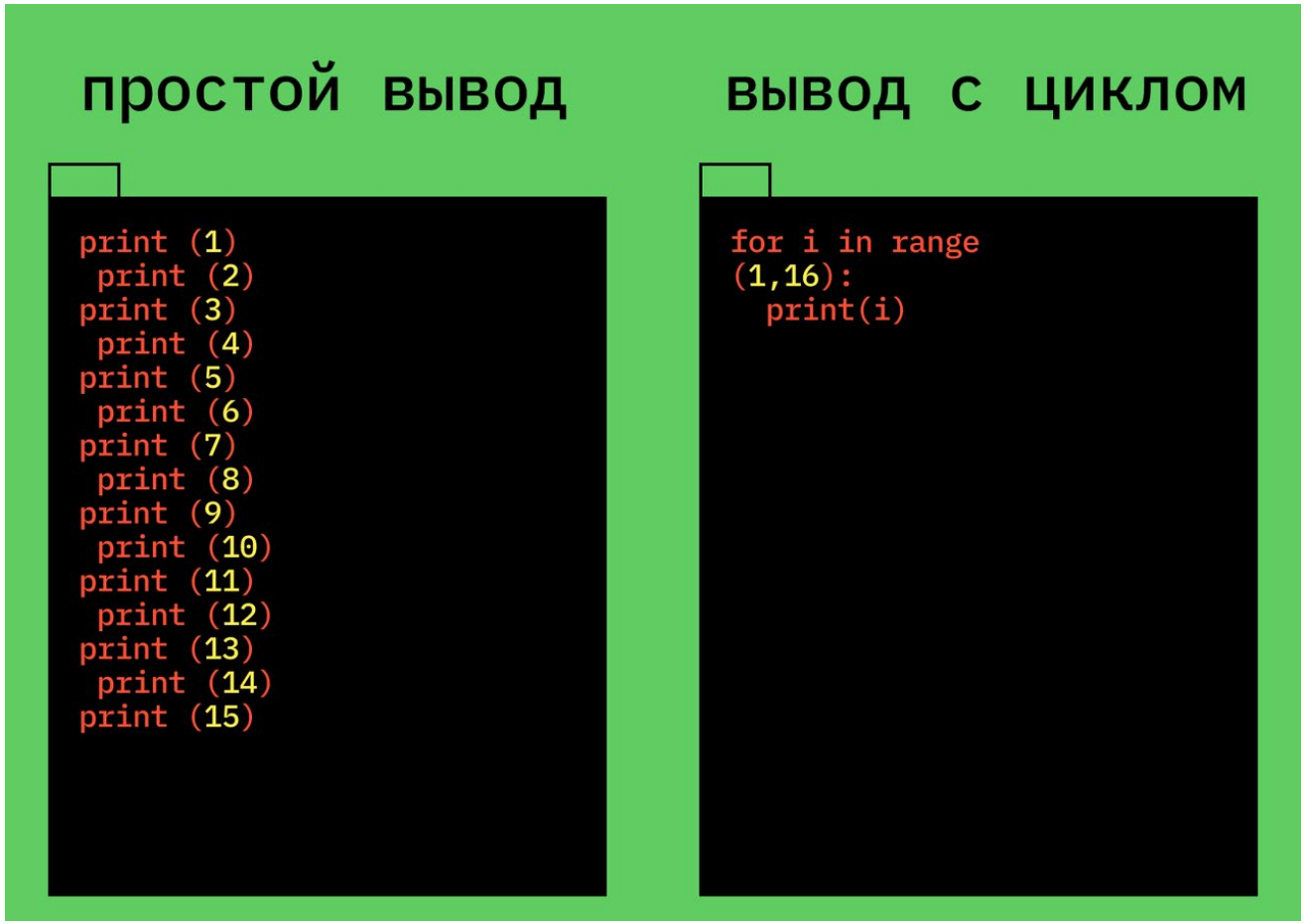
```
>>> number = -10
>>> abs_number = number if number >= 0 else -number
>>> print(abs_number)
10
```



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Циклы

Циклы в языке Python представлены двумя основными конструкциями: **while** и **for**.



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Цикл while

Цикл while (“пока”) позволяет выполнить одну и ту же последовательность действий, пока проверяемое условие истинно.

Синтаксис цикла while:

```
while условие:
    блок инструкций
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Цикл while

```
1    number = 1
2    while number < 5:
3        print(f"Число = {number}")
4        number += 1
5    print("Работа программы завершена")
```

```
Число = 1
Число = 2
Число = 3
Число = 4
Работа программы завершена
```

Тело цикла выполнилось 4 раза

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Цикл for

Цикл for пробегается по набору значений, помещает каждое значение в переменную, и затем в цикле мы можем с этой переменной производить различные действия.

Синтаксис цикла for:

for переменная in набор_значений:

    инструкции

```
1   friends = ['Андрей', 'Максим', 'Ирина']
2   for friend in friends:
3       print('Привет,', friend)
4   print('Готово!')
```

```
Привет, Андрей
Привет, Максим
Привет, Ирина
Готово!
```

СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Функция range

Для получения последовательности чисел можно воспользоваться функцией **range**:

- range(stop);
- range(start, stop);
- range(start, stop, step).

```
# 0 - начальный элемент по умолчанию
for a in range(3):
    print(a)
```

```
0
1
2
```

```
# два аргумента
for b in range(7, 10):
    print(b)
```

```
7
8
9
```

```
# три аргумента
for c in range(1, 13, 3):
    print(c)
```

```
1
4
7
10
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Управление циклом

Оператор **break** осуществляет выход из цикла.

А оператор **continue** выполняет переход к следующей итерации цикла.

```
number = 0
while number < 5:
    number += 1
    if number == 3: # если number = 3, выходим из цикла
        break
    print(f"Число = {number}")
```

```
Число = 1
Число = 2
```

```
number = 0
while number < 5:
    number += 1
    if number == 3: # если number = 3, переходим к новой итерации цикла
        continue
    print(f"Число = {number}")
```

```
Число = 1
Число = 2
Число = 4
Число = 5
```

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Вложенные циклы

Внешний цикл

Вложенный цикл

```
for i in range(1, 10):
    for j in range(1, 10):
        print(i * j, end="\t")
    print()
```

Тело вложенного цикла

Тело внешнего цикла

```
1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81
```