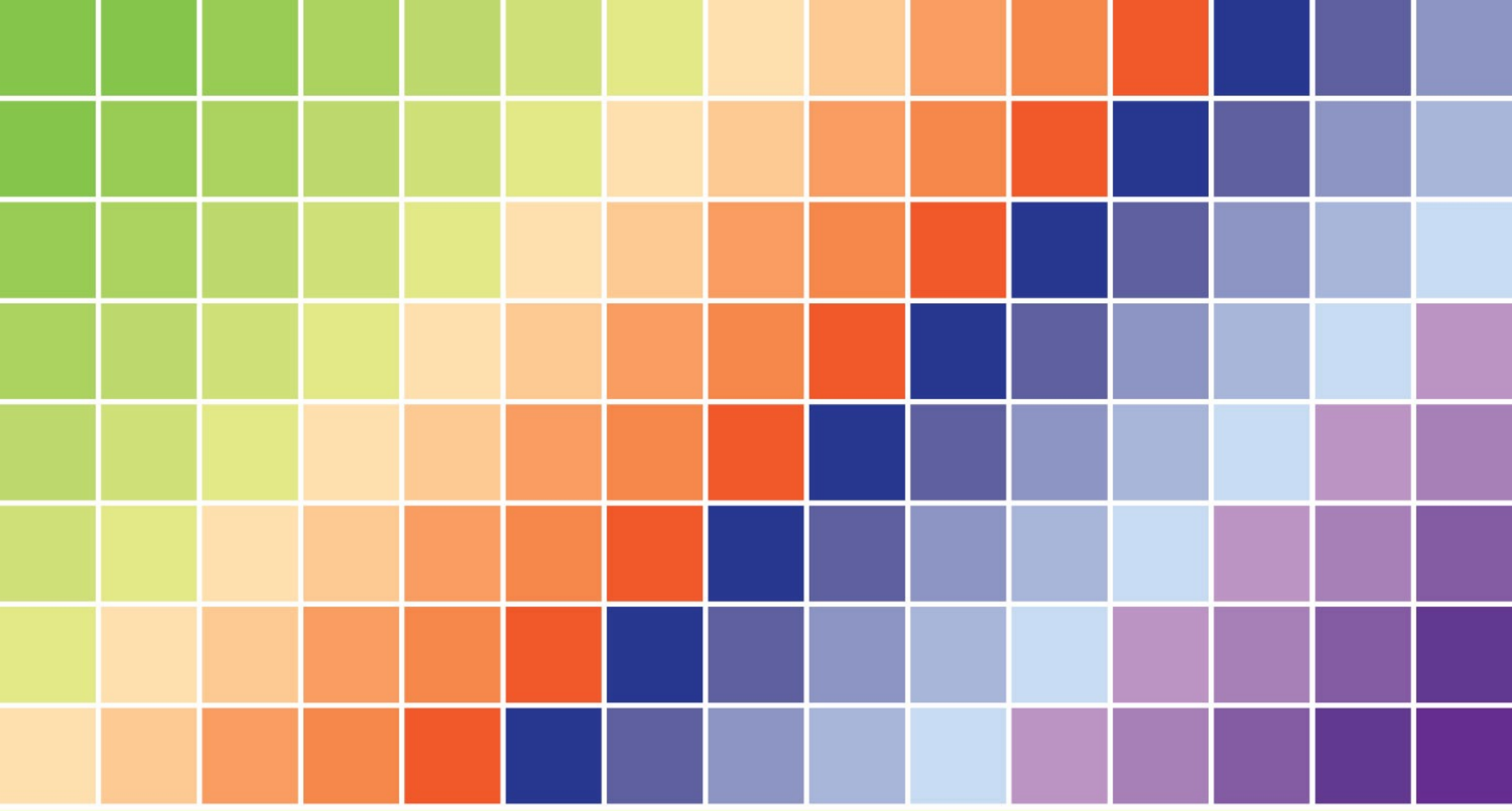


School of Computer Science

КОРТЕЖИ. СОБСТВЕННЫЕ ФУНКЦИИ

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

– Выберите из списка ниже случаи, подходящие для использования цикла while и цикла for в Python:

1. Количество итераций зависит от ввода пользователя
2. Конечное, заранее известное количество итераций
3. Бесконечный цикл с выходом по условию в теле цикла с помощью break
4. Перебор значений из списка

?

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

Рассмотрите этот код:

```
for <элемент> in <список>:
    <блок1>             # может ли <блок1> НЕ исполниться?
    if <условие>:       # и в каком случае?
        continue
    else:
        break
    <блок2>             # а <блок2> когда-нибудь выполнится?
else:
    <блок3>             # когда выполнится <блок3>?
```

Ваши ответы?

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

– Какой ряд чисел образуется после выполнения алгоритма

```
for i in range(1,10+1):
    print(i):
```

1) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
2) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
3) 1, 11, 23, 36, 40, 45, 61, 78, 96, 115

?

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

- Рассмотрите код, который печатает с шагом 3 символы строки str_3:

```
>>> str_3 = 'abcABCabcABC'
>>> for i in range(0, len(str_3), 3):
        print(_______)
```

- Что должен напечатать этот код?
- Выберите выражения, которые можно подставить в функцию print():
  1. i
  2. str_3[i : i+3]
  3. str_3[i + i]
  4. str_3[i]

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

– К чему приведет обращение к непустому списку по индексу «-1»:

1. Вернётся последний элемент
2. Ошибка IndexError
3. Вернется первый элемент
4. Ошибка KeyError

?

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

- Этот двумерный список представляет собой матрицу 3x3:

```
>>> table = [[1,2,3], [4,5,6], [7,8,9]]
```

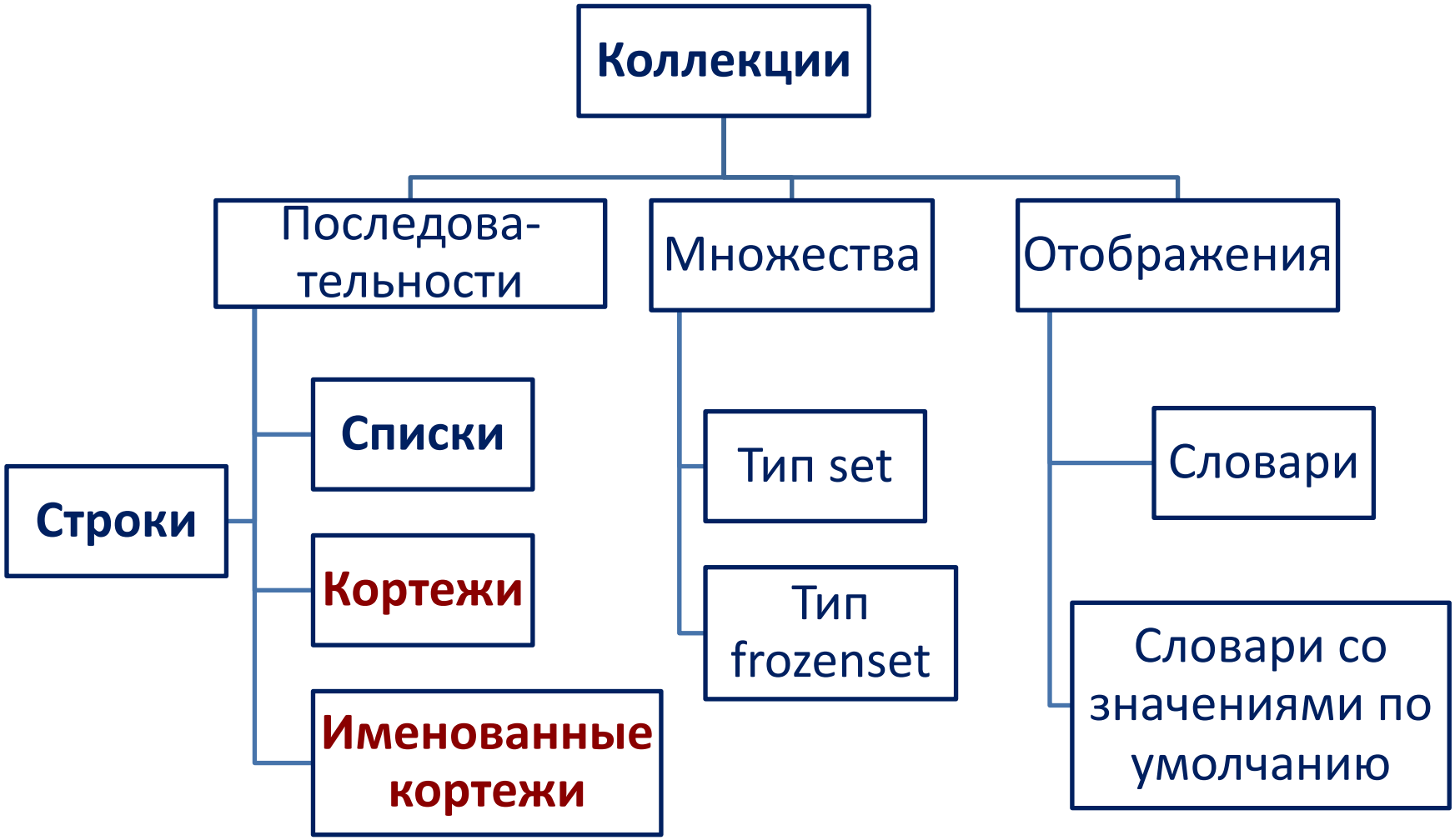
- Как вывести элемент в середине главной диагонали этой матрицы?
- Выберите подходящее выражение:
  1. table[2]
  2. table[1]
  3. table[1, 1]
  4. table[1][1]

ВОПРОС ПО ПРОШЛОМУ ЗАНЯТИЮ

- Имеется программа, код которой указан ниже.
- Укажите, какие значения будут содержать списки в помеченных участках:

```
1    list_1 = [1, 2, 3]
2    list_2 = list_1
3    # значения списка list_2?
4
5    list_1[1] = 10
6    # значения списка list_2?
7
8    list_2[0] = 20
9    # значения списка list_1?
10
11   list_1 = [5, 6]
12   # значения списка list_2?
13
```

ТИПЫ КОЛЛЕКЦИЙ В PYTHON



КОРТЕЖИ (TUPLES)

- Кортеж – это последовательность, поддерживающая следующие операции:
  - Конкатенация и тиражирование: + и *
  - Проверки на вхождение in и not in
  - Функция определения размера len(object)
  - Индексация: object[index]
  - Извлечение срезов: object[start:stop:step]
  - Итерации, гарантирующие строгую последовательность элементов
- Соответствует строке данных в таблице

- Кортеж, в отличие от списков, это **неизменяемый** тип данных

ИНИЦИАЦИЯ КОРТЕЖЕЙ

- Объявление:
  - tuple()
  - ()
  - (*элементы через запятую*)

- Примеры:
  - `tup1 = tuple()` # пустой кортеж
  - `tup2 = ()` # тоже пустой кортеж
  - `tup3 = (1,)` # кортеж с одним #элементом
  - `tup3 = (-17.5, 'text', 83)`
  - `tup4 = (['Monday', -18], # кортеж ['Tuesday', +5], # с вложен- 'Unknown' ) # ными списками`

МЕТОДЫ КОРТЕЖЕЙ

Вызов	Описание
T.count(x)	Возвращает число вхождений элемента x в кортеж T
T.index(x [, start, end])	Возвращает индекс самого первого (слева) вхождения элемента x в кортеж T (или в срез start:end кортежа T) или возбуждает исключение ValueError

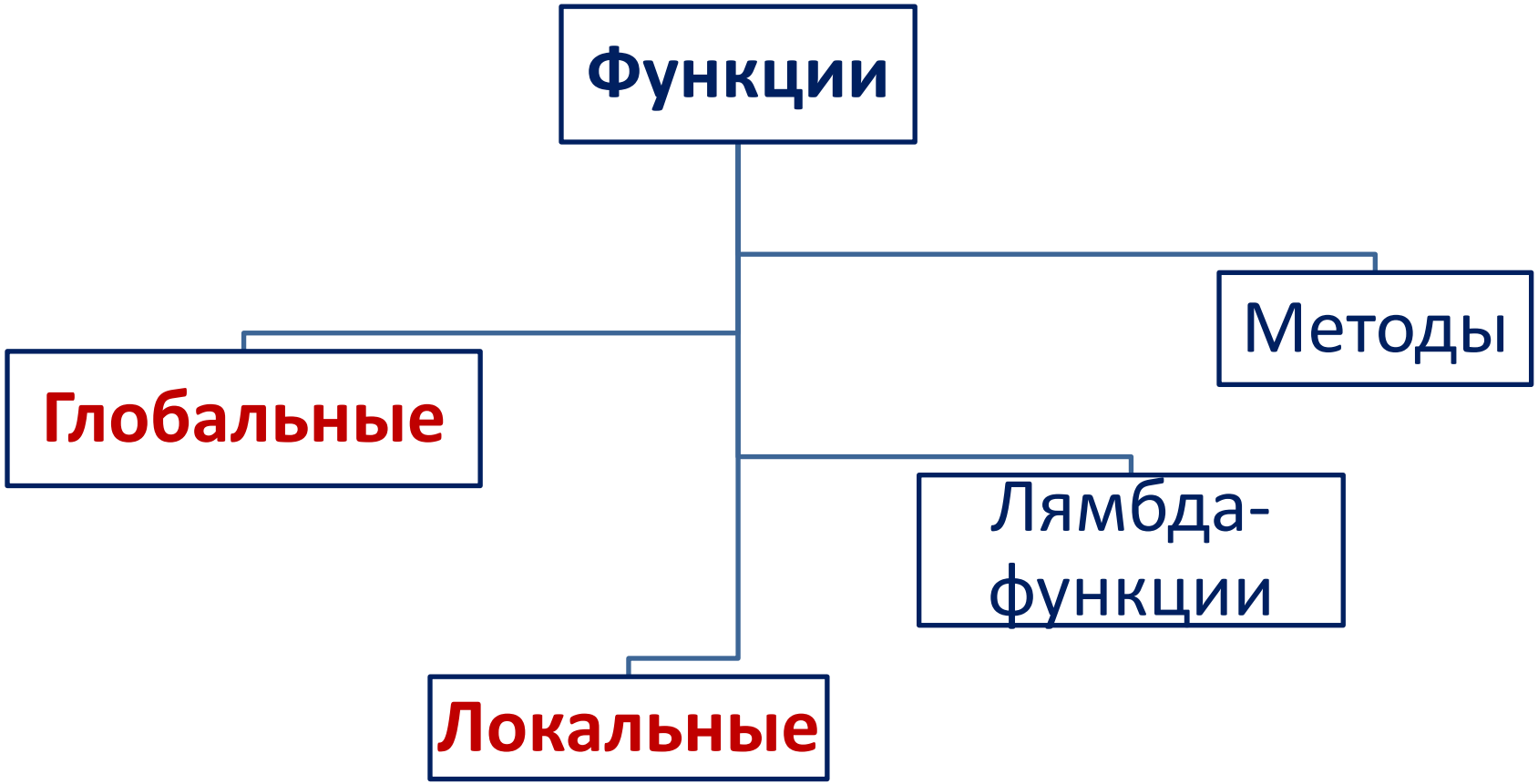
ПРИМЕРЫ РАБОТЫ С КОРТЕЖАМИ

```
>>> tup1 = (1, 2, 3, 4)      # кортеж из 4-ех элементов
>>> len(tup1)                # длина
4
>>> tup1 + (5, 6)            # конкатенация
(1, 2, 3, 4, 5, 6)
>>> tup1[1]                  # извлечение элемента
2
>>> tup1[2:4]                # извлечение среза
(3, 4)
>>> tup1.index(4)            # значение 4 находится в позиции 3
3
>>> tup1.count(4)            # значение 4 присутствует 1 раз
1
>>> tup1[0] = 0              # кортежи являются неизменяемыми
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    tup1[0] = 0             # кортежи являются неизменяемыми
TypeError: 'tuple' object does not support item assignment
```

ИМЕНОВАННЫЕ КОРТЕЖИ

- Инициируются с помощью collections.namedtuple()
- Нужен предварительный import collections
- Позволяет адресоваться к элементам кортежа не только числовыми, но и текстовыми индексами, например:
  - sale.quantity * sale.price
  - aircraft.seating.maximum
- См. примеры в tuple_indexing.py и tuple_named.py

ВИДЫ ФУНКЦИЙ В PYTHON



ФУНКЦИИ. СИНТАКСИС

Описание:

Отступы обяза-тельны!

def <имя>([формальные параметры]):
⟷ <*код функции*>
[**return** x]

Вызов: <имя>([аргументы])

Функция без **return** возвращает **None**

ПЕРЕДАЧА ПАРАМЕТРОВ И ВОЗВРАТ ЗНАЧЕНИЙ

- Смотрите примеры в receive_and_return.py

- Именованные и позиционные параметры, значения по умолчанию:
  - Пример в birthday_wishes.py
  - Параметры со значениями по умолчанию должны объявляться последними
  - При вызове функции позиционные параметры должны передаваться в первую очередь

ПУСТЫЕ ФУНКЦИИ

```
def empty_func():
    pass
```

- Этот подход используется для того, чтобы отложить разработку кода на будущее
- Оператор pass можно использовать и в других блоках кода

DOC STRING

```
def empty_func():
    """ Describe function profile

        and how it works,
            ...
    """
    pass
```

ТИПИЧНЫЙ СОСТАВ DOC STRING

1. Типы параметров и возвращаемых значений
2. Описание того, что делает функция
3. Условия ее использования
4. Возбуждаемые исключения (если есть)
5. Примеры вызовов в стиле Shell

- Смотрите примеры в triangle.py

- Для пояснений хитрого алгоритма комментарии внутри исходного кода более предпочтительны, чем строки документации

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

- Рассмотрите библиотеку triangle.py
- Напишите функцию hypotenuse() в модуль triangle.py по скрипту NonFunc\hypot_len_v2.py
- Требования :
  - Функция должна принимать 2 параметра – положительные длины катетов
  - В функции должен присутствовать Doc String, выдающий справку в среде разработки

ДОМАШНЕЕ ЗАДАНИЕ. БИБЛИОТЕКА CIRCLE

- Создайте свою библиотеку circle.py
- В ней должны быть функции:
  - area(radius) – площадь круга
  - sector_area_len(radius, length) – площадь сектора круга через длину дуги
  - sector_area_ang(radius, angle) – площадь сектора круга через угол
  - circumference(radius) – длина окружности
  - volume(radius, height) – объём цилиндра
- Заполнение Doc String – обязательно для модуля и для каждой функции
- Значение числа Pi смотрите в math.pi

СПАСИБО ЗА ВНИМАНИЕ !

ВОПРОСЫ ?

School of Computer Science