



Вводное занятие

Докладчик: Евграфов Михаил

Организационные вопросы

Коммуникация



ТГ-канал курса



Репозиторий курса

Система оценивания

$$grade_{final} = \frac{2 \times grade_{theory} \times grade_{practice}}{grade_{theory} + grade_{practice}}$$

$grade_{final}$ - **финальная оценка**

$grade_{theory}$ - **оценка за устный зачет**

$grade_{practice}$ - **оценка за работу в семестре**

Оценка за работу в семестре

$$grade_{practice} = 0.6 \times grade_{hw} + 0.4 \times grade_{sem}$$

$grade_{hw}$ - средняя оценка за домашки

$grade_{sem}$ - оценка за работу на семинарах

Как оценивается работа на семинарах

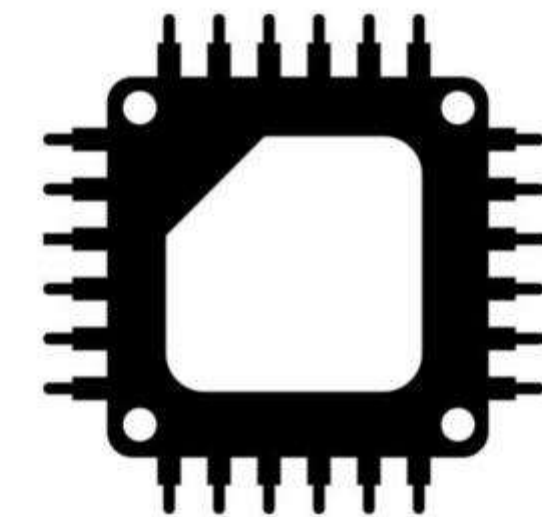
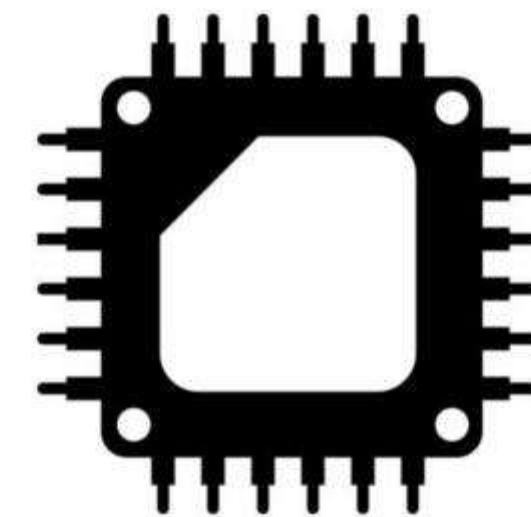
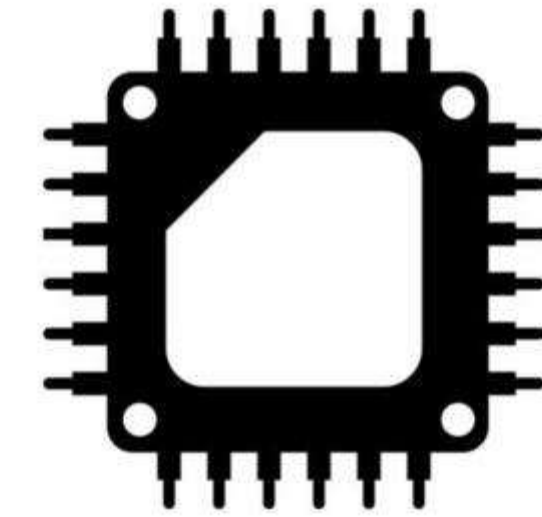
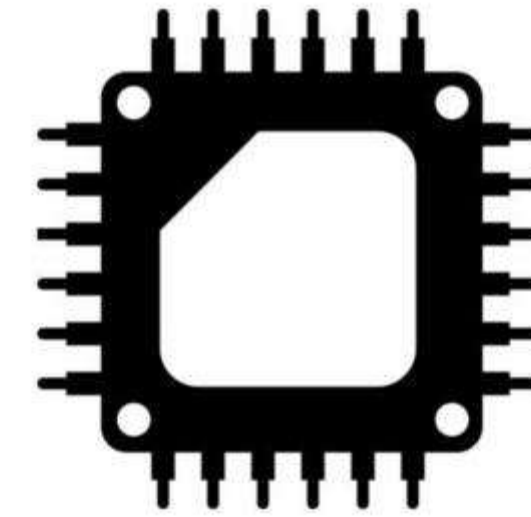
- Каждый семинар решаем задачи
- Решения, сданные в день семинара, оцениваются на полный балл
- Решения, сданные в течении недели после семинара, оцениваются с коэффициентом 0.2
- По прошествии недели решения оцениваются 0
- Результаты списавших обнуляются
- Результаты людей, давших списать, засчитываются с коэффициентом 0.5

Введение в Python

Язык и реализация

Язык программирования

```
if __name__ == "__main__":  
    print("Hello world")
```



Реализация

Реализации Python



ironpython



ironpyth



Jython



pypy




nv/nv



Какой язык Python?

**Python - это мультипарадигмальный
высокоуровневый язык
программирования общего назначения с
динамической строгой типизацией и
автоматическим управлением памятью**

Мультипарадигмальный

Python - это мультипарадигмальный
высокоуровневый язык
программирования общего назначения с
динамической строгой типизацией и
автоматическим управлением памятью

Python и ООП

```
@dataclass
class Vector2D:
    x: float = 0
    y: float = 0

    def __abs__(self) -> float:
        return (self.x ** 2 + self.y ** 2) ** 0.5
```

```
point = Vector2D(x=42)
abs(point)
```


Функциональный Python

```
def get_composition(  
    func1: Callable,  
    func2: Callable,  
) -> Callable:  
    return lambda x: func1(func2(x))
```

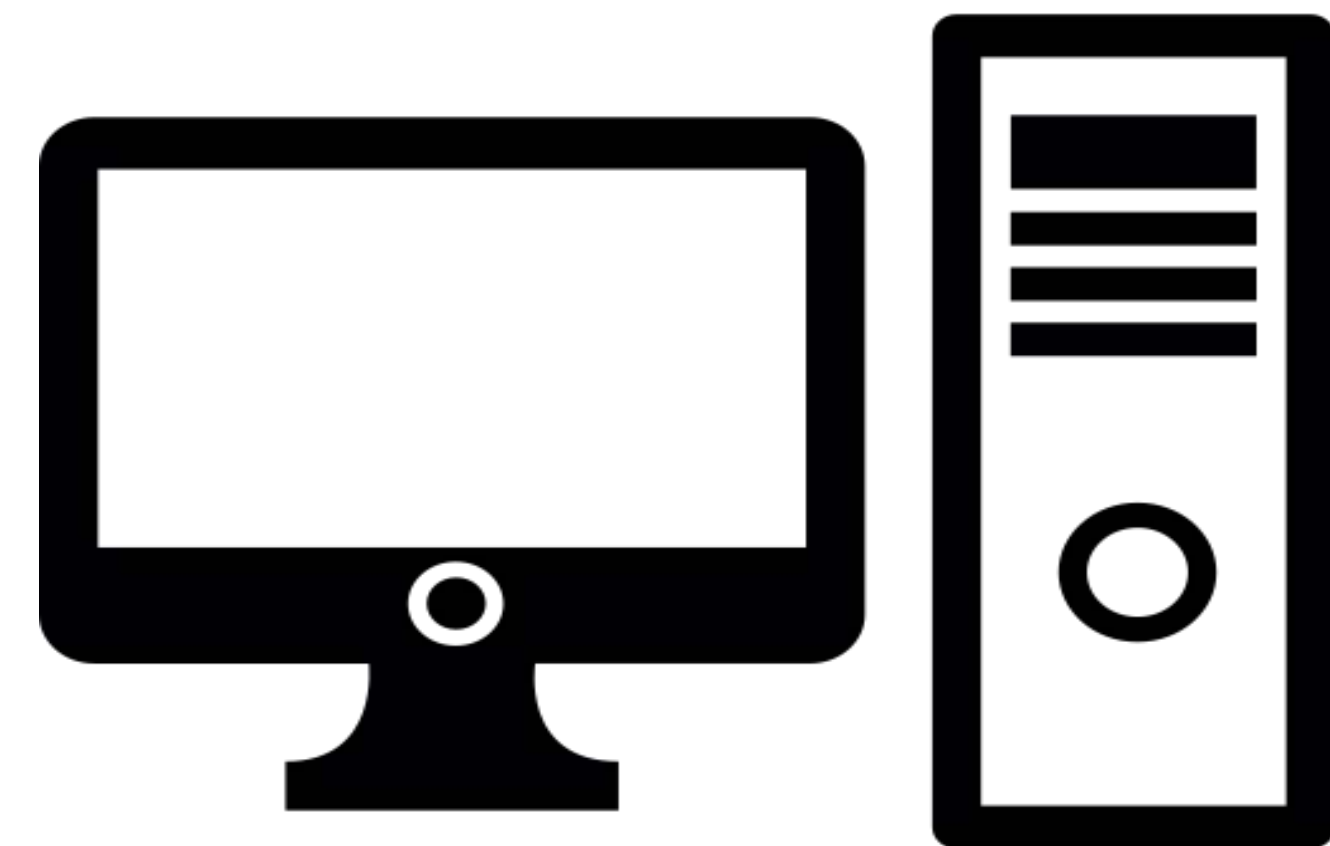
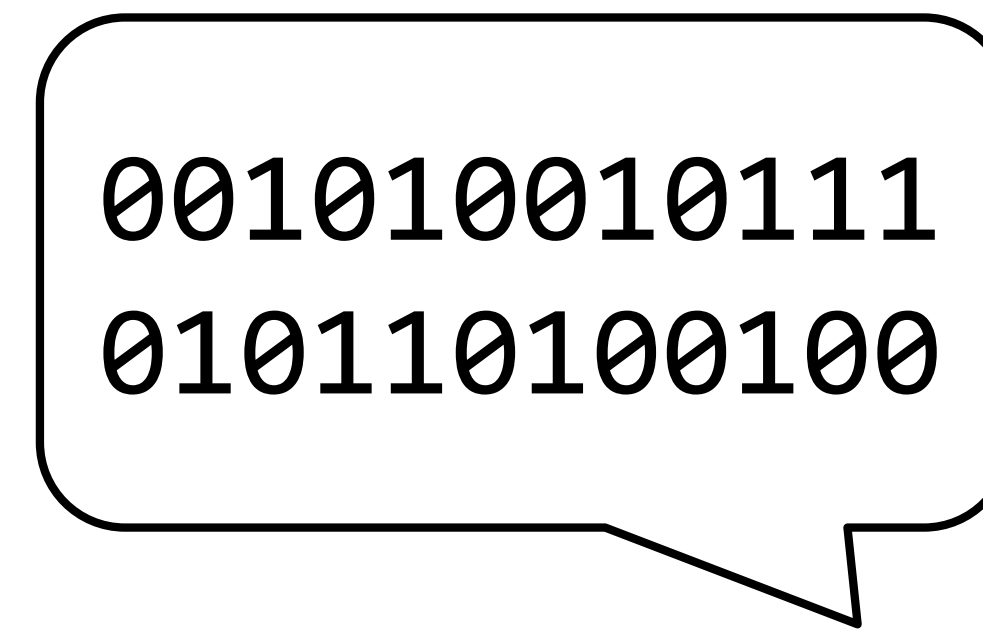
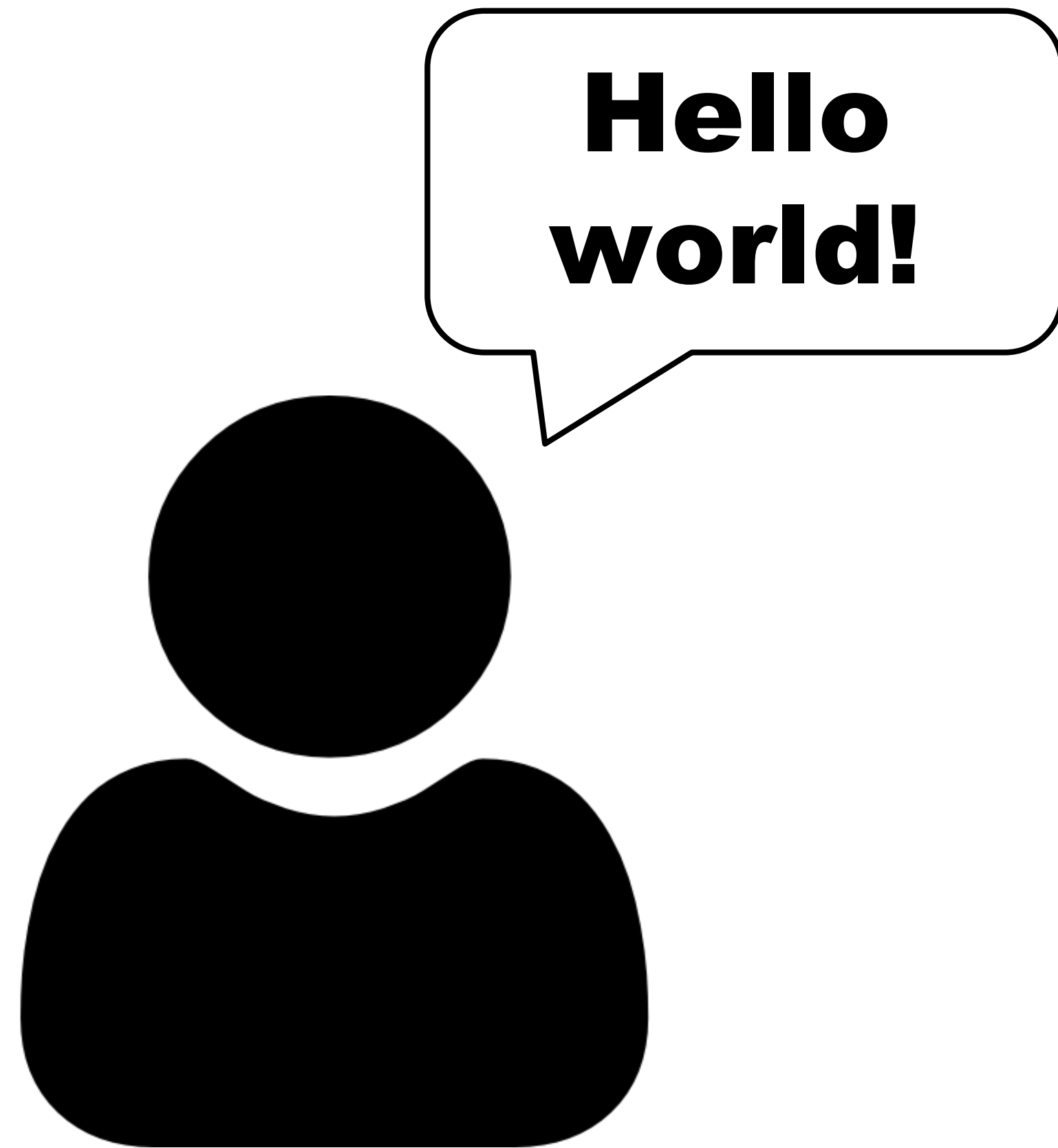
```
>>> get_composition(  
    lambda x: x * 2,  
    lambda x: x ** 2,  
) (3)
```

18

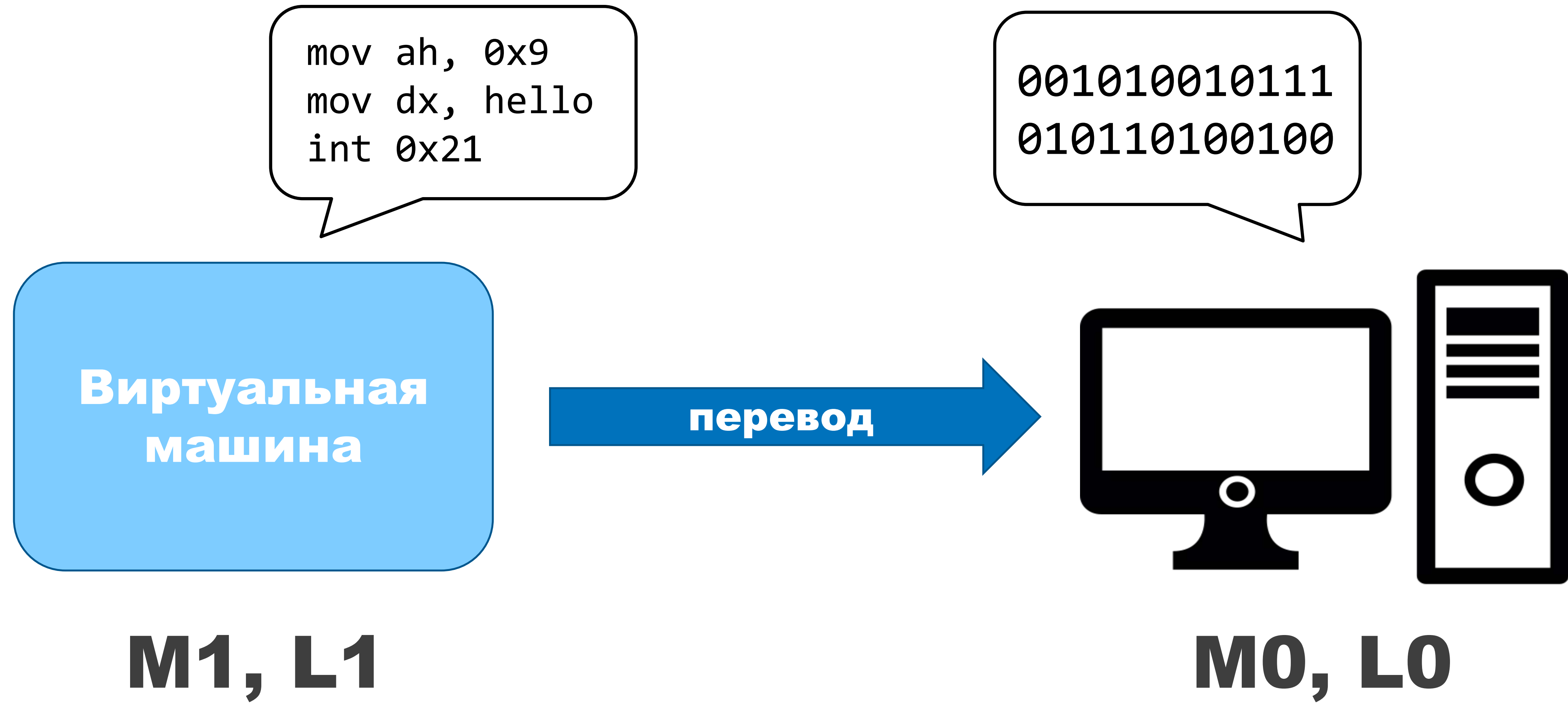
Высокоуровневый

Python - это мультипарадигмальный
высокоуровневый язык
программирования общего назначения с
динамической строгой типизацией и
автоматическим управлением памятью

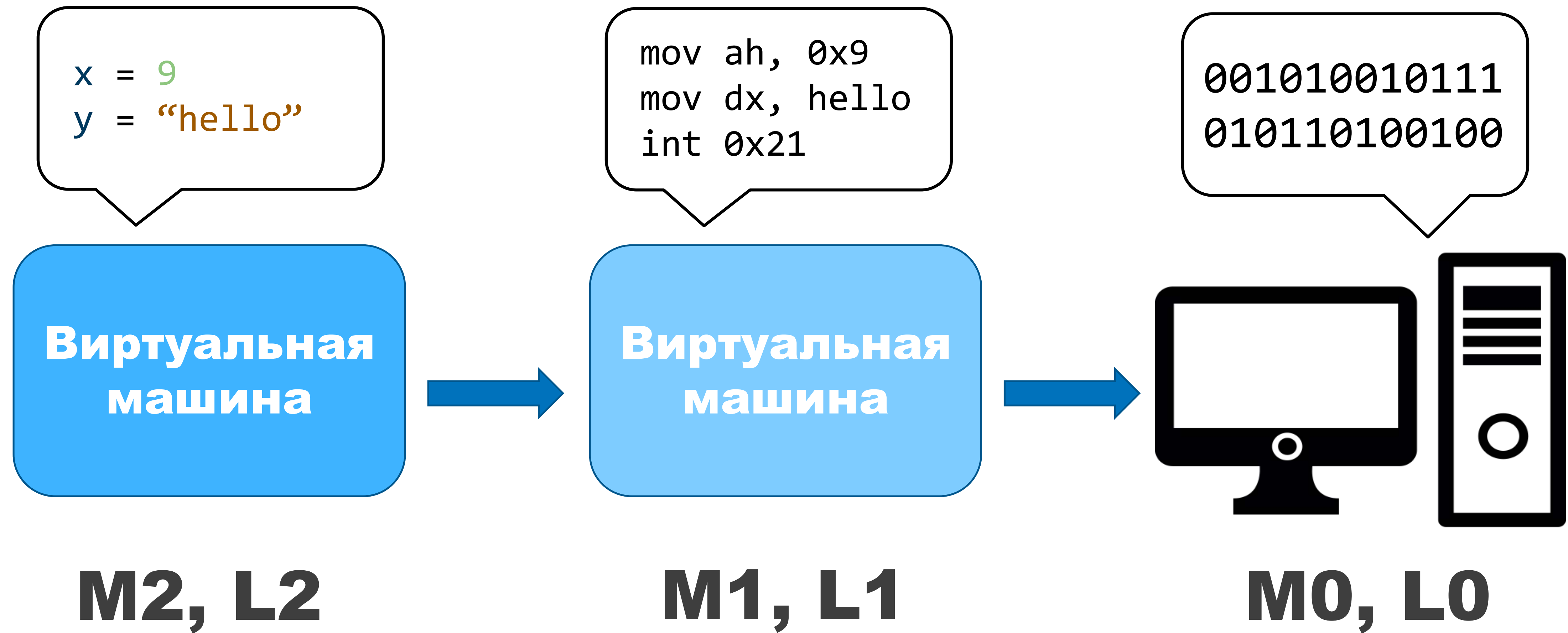
Люди и компьютеры



Виртуальные машины

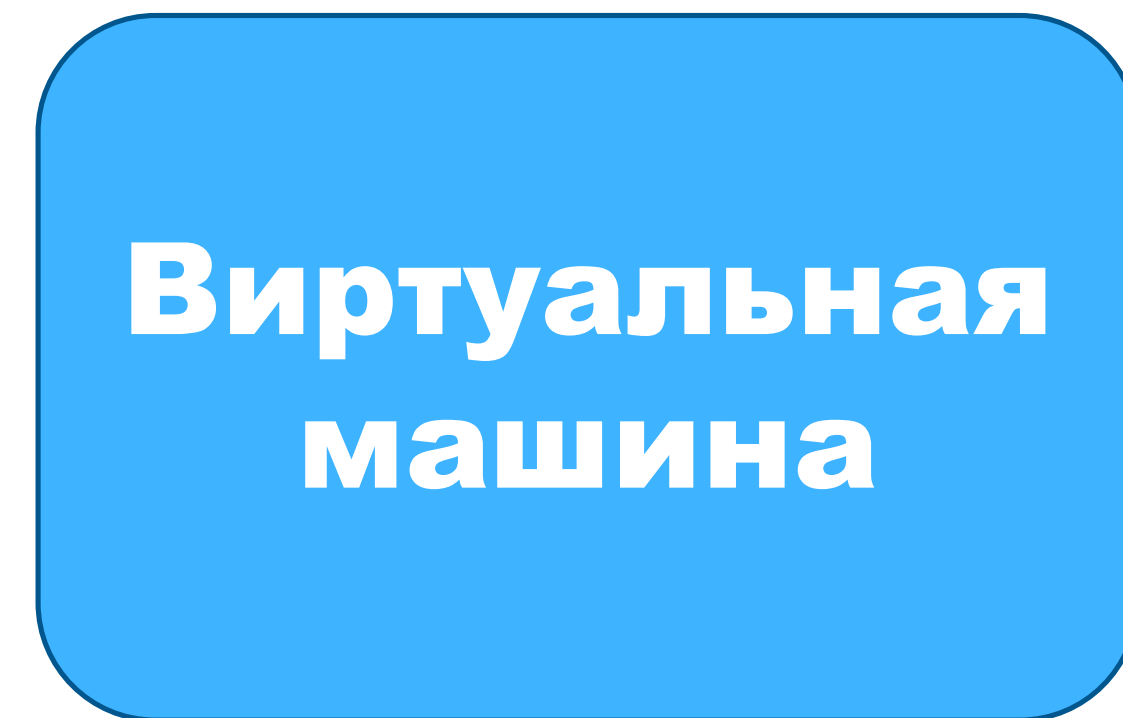


Языки разных уровней



Языки разных уровней

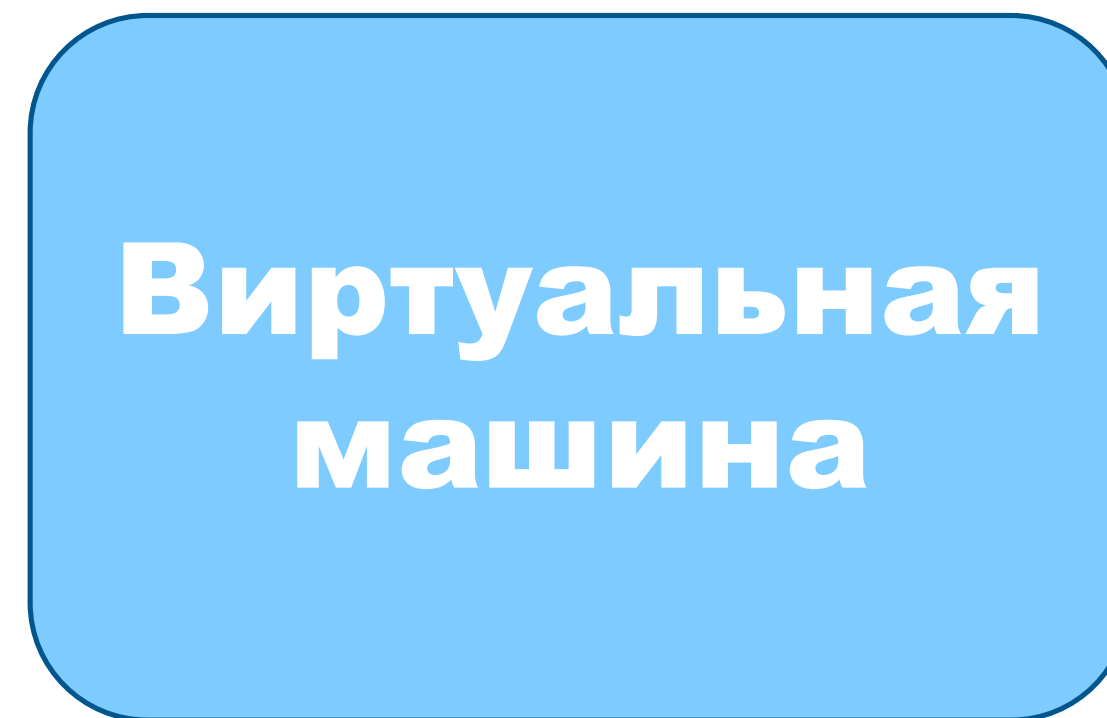
**ЯЗЫК ВЫСОКОГО
уровня**



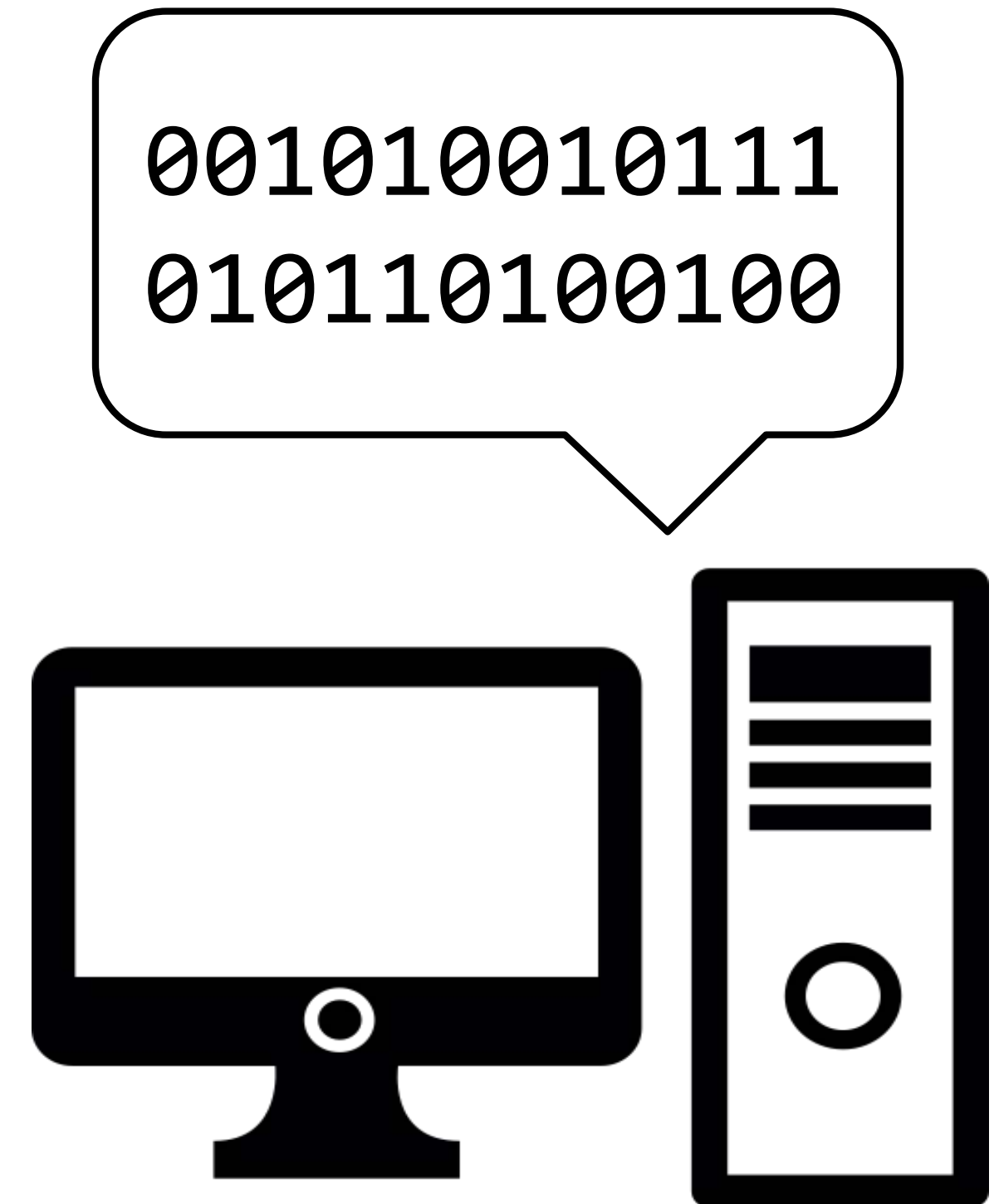
M2, L2



**ЯЗЫК НИЗКОГО
уровня**



M1, L1



M0, L0

Высокоуровневость Python

```
banned_elems = set()
```

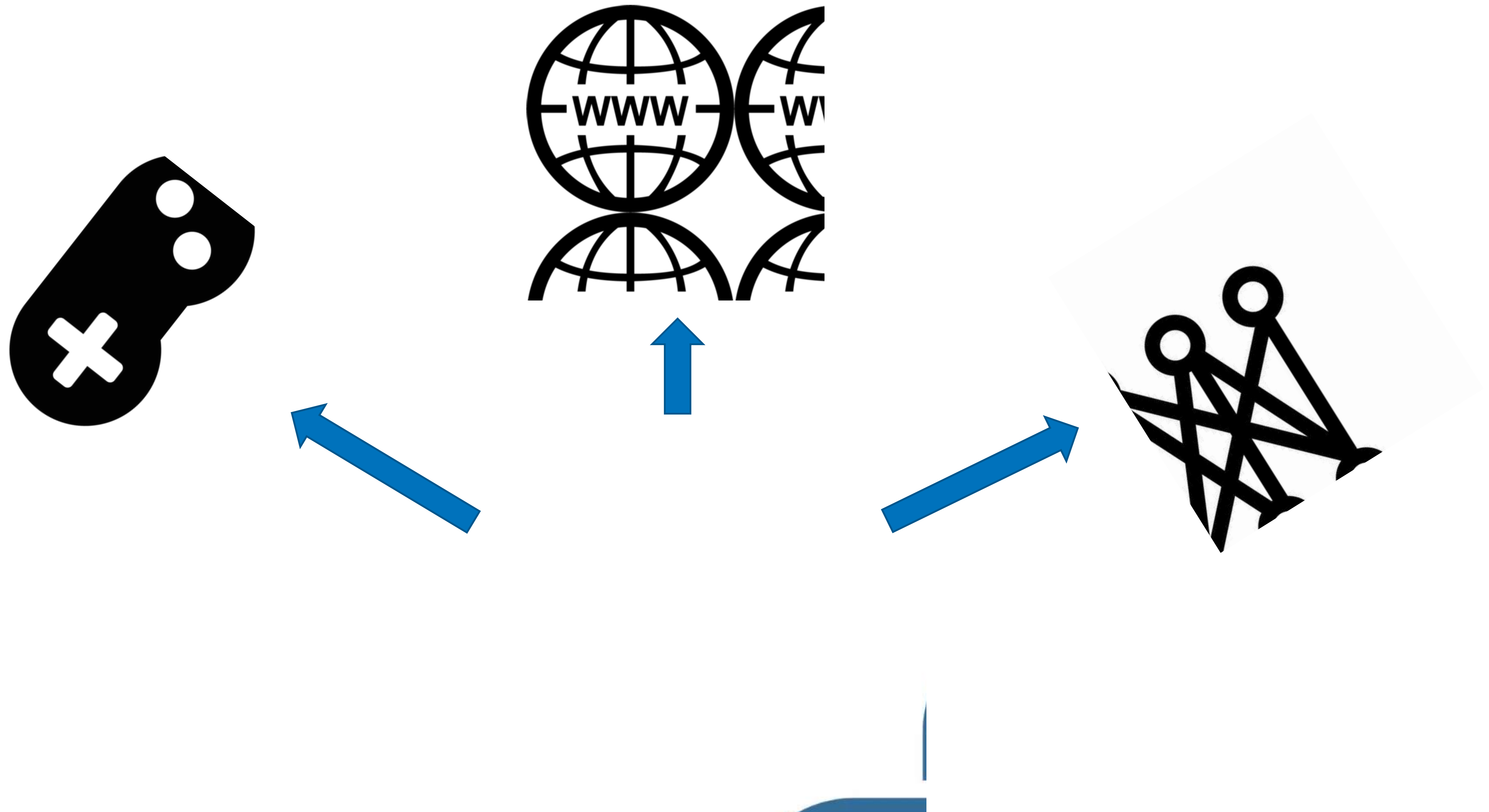
```
if elem not in banned_elems:
```

```
    ...
```

Язык общего назначения

Python - это мультипарадигмальный
высокоуровневый **язык**
программирования общего назначения с
динамической строгой типизацией и
автоматическим управлением памятью

Язык общего назначения



Язык с динамической типизацией

Python - это мультипарадигмальный
высокоуровневый **язык**
программирования общего назначения с
динамической строгой **типизацией** и
автоматическим управлением памятью

Виды типизации

- **Статическая - проверка типов «до»**
- **Динамическая - проверка типов «во время»**

Язык со строгой типизацией

Python - это мультипарадигмальный
высокоуровневый **язык**
программирования общего назначения с
динамической **строгой типизацией** и
автоматическим управлением памятью

Виды типизации

- **Нестрогая - частые неявные преобразования**

```
>>> "banana" + 42 # OK
```

- **Строгая - почти нет неявных преобразований**

```
>>> "banana" + 42 # Error
```

Градиенты типизации

Статическая

Динамическая



Строгая

Нестрогая



Управление памятью

Python - это мультипарадигмальный
высокоуровневый **язык**
программирования общего назначения с
динамической строгой типизацией и
автоматическим управлением памятью

Объекты

```
struct _longobject {  
    long ob_refcnt;  
    PyTypeObject *ob_type;  
    size_t ob_size;  
    long ob_digit[1];  
}
```


Сбор мусора

```
struct _longobject {  
    long ob_refcnt; // 2  
    PyTypeObject *ob_type;  
    size_t ob_size;  
    long ob_digit[1];  
}
```

num1



```
graph LR; num1[num1] --> ob_refcnt[ob_refcnt]; num2[num2] --> ob_digit[ob_digit]
```

num2

Сбор мусора

```
struct _longobject {  
    long ob_refcnt; // 1  
    PyTypeObject *ob_type;  
    size_t ob_size;  
    long ob_digit[1];  
}
```



num1

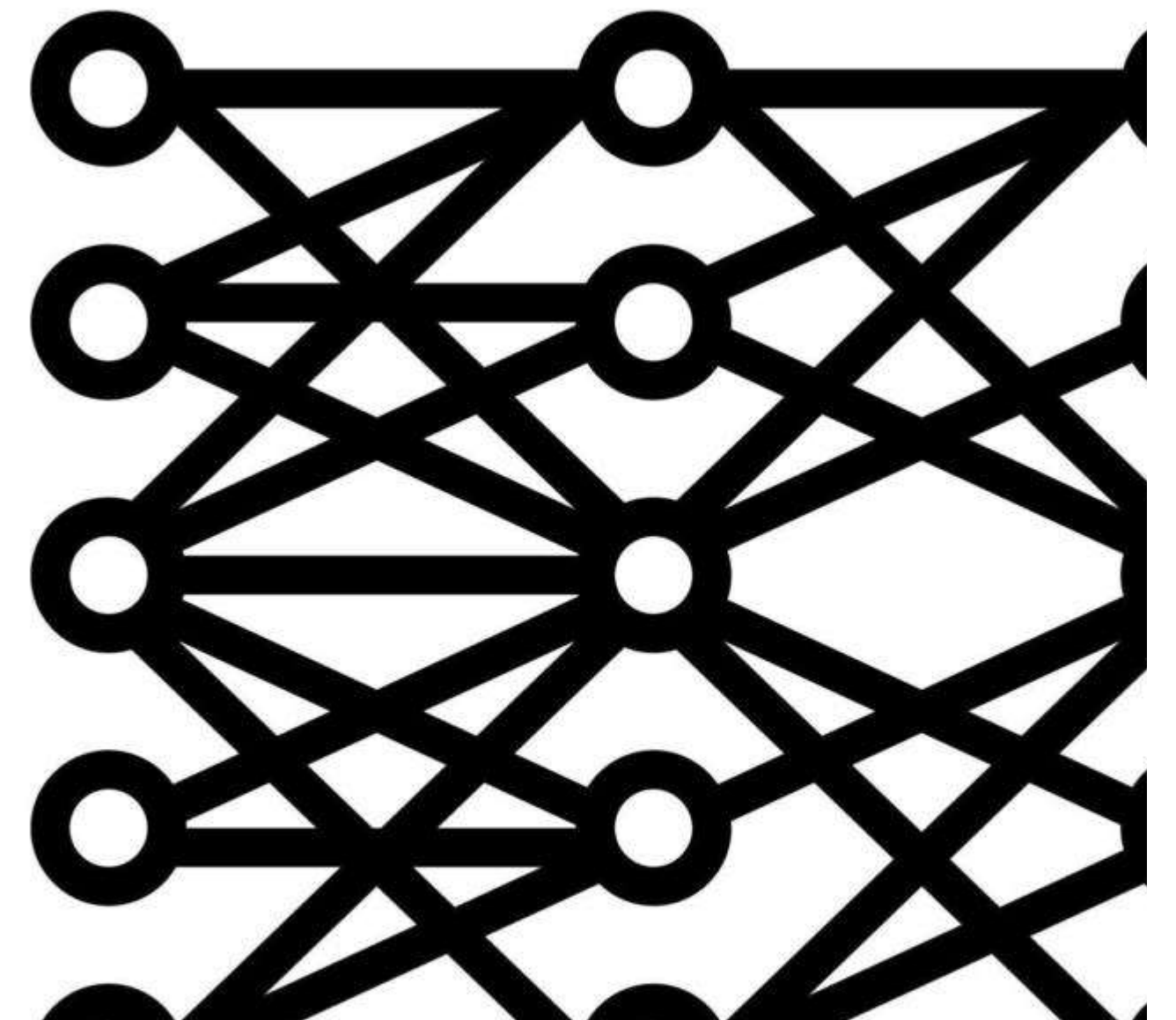
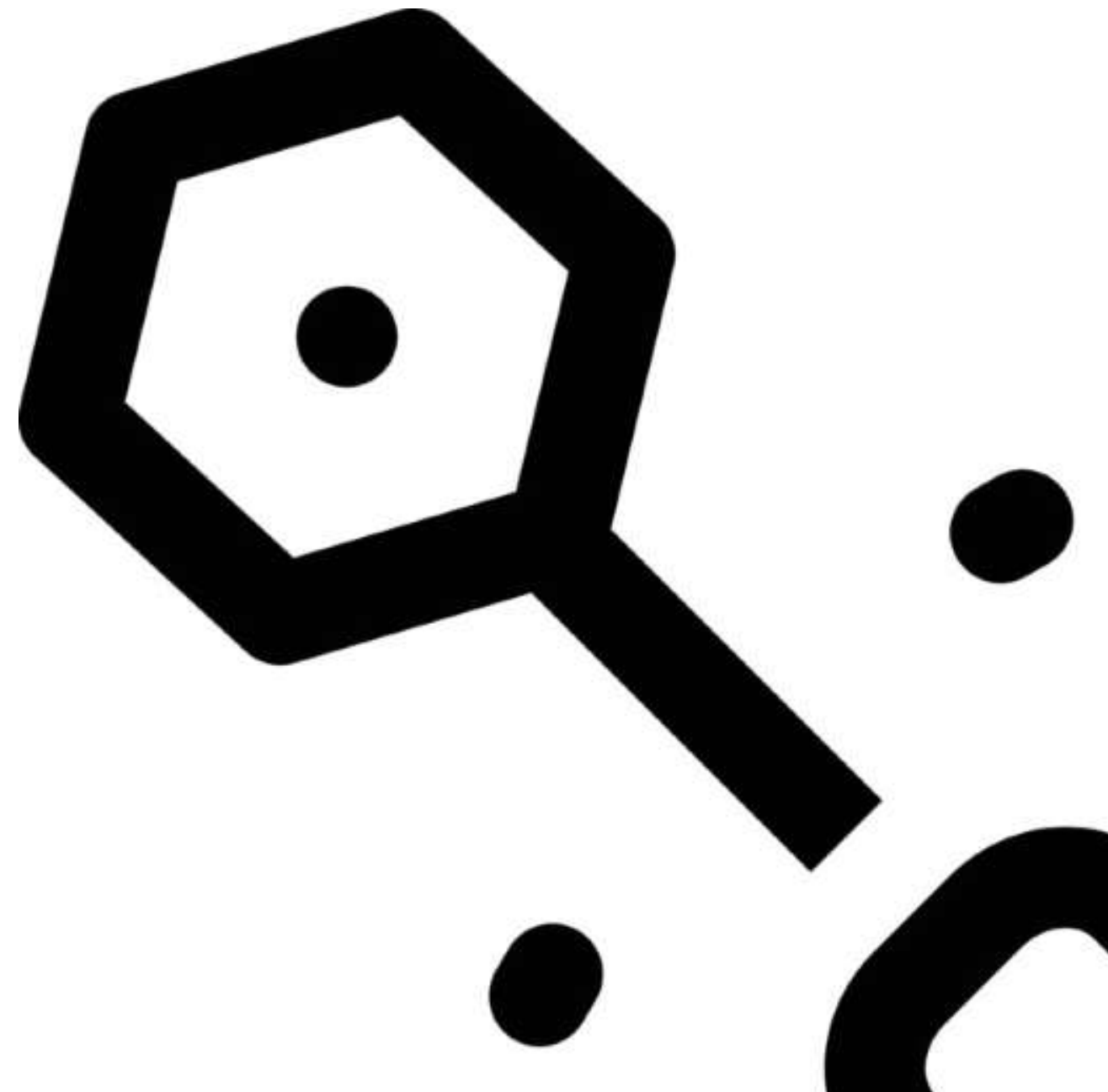
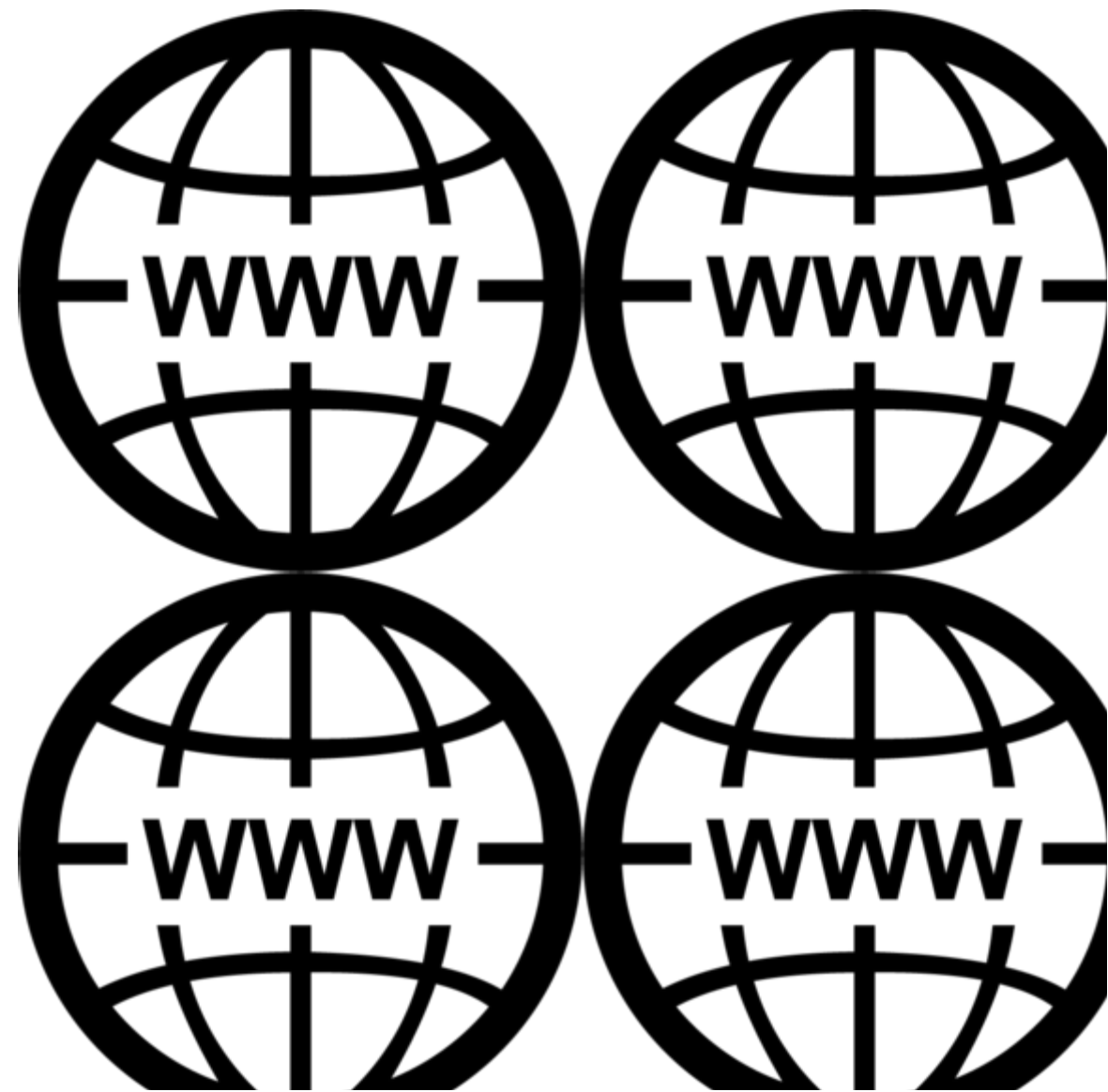
A blue rounded rectangle labeled 'num1' has a blue arrow pointing from its left side to the 'ob_refcnt' field in the C struct definition above.

Сбор мусора



```
struct _ngobject {  
    long int; // 0  
    PyTypeObject *ob_type;  
    size_t;  
    1; ob_data[];  
}
```

Что пишут на Python?



Семинар