

## Описание кода:

### 1. Словарь символов для набора

Задаёт соответствие индексов буквам: `0..9 -> 'a'..'k' (без 'h')`. Печатаёт ключи и значения. Это набор целевых классов (10 букв).

### 2. Генерация синтетических ERP/EEG-подобных данных

- Определяет две волны:
  - `p300\_wave`: синус с фазой, сдвинутый вверх (+1) — имитация положительной P300.
  - `n100\_wave`: масштабированный отрицательный синус — имитация раннего компонента N100.
- Параметры выборки:
  - `n\_trials` = 10` на букву, `n\_letters` = 10` букв, `n\_ch` = 8` каналов.
- Для каждой буквы и каждого прогона:
  - Случайная длина нулевого промежутка до `n\_before` (90–150 tps) и после `n\_after` так, чтобы общая длина сегмента была 400 tps.
  - Формирует сигнал: нули до + `n100\_wave` + `p300\_wave` + нули после.
  - Добавляет гауссов шум ( $\sigma=0.3$ ).
  - Канал 0 получает полный сигнал; каналы 1–7 получают тот же сигнал с случайной амплитудой (0.1–0.4) и собственным шумом — имитация разных амплитуд по электродам.
  - Сохраняет как массив формы (8, 400) на один пример.
  - В `targets` кладёт пару [буква, номер\_прогона].
- На выходе `dataset\_simul` имеет форму `(100, 8, 400)` — 100 примеров (10 букв × 10 прогонов), 8 каналов, 400 временных точек.

### 3. Визуализация спектрограммы амплитуд

Отображает `imshow` модуля значений объединённого по времени набора (все примеры склеены последовательно), участок временной оси `[10000:12000]`, вид: каналы × время. Это позволяет глазом оценить «всплески» ERP и шум.

### 4. Формирование длительной последовательности с несколькими «взглядами» на букву

**Идея:** собрать длинный непрерывный сигнал из нескольких выбранных эргономичных «эпизодов» (гейзов) для разных букв, разделённых случайными промежутками шума.

- Создает список целевых букв `targets2type` длиной 30 (каждая буква по 3 раза), перемешивает.
- Для каждой буквы в этом списке:
  - Случайно выбирает `n\_gazes` от 3 до 6 — число эпизодов для этой буквы.

- Находит индексы всех примеров данной буквы в `targets`, случайно выбирает нужное число примеров из `dataset\_simul`.
- Для каждого выбранного примера добавляет справа случайный шумовой разрыв по времени длиной `30..60` тпс (шум  $\sim N(0, 0.3)$  на каждый канал).
- Конкатенирует выбранные и «расширенные» эпизоды по времени в один блок для текущей буквы.
- Затем конкатенирует все блоки для 30 буквенных «заданий» по времени в одну большую матрицу `new\_dataset\_simul` формы `(8, ~60k)` — 8 каналов, ~62k временных точек. Это имитирует непрерывную запись, где периодически появляются ERP-ответы на множество «взглядов» на символы, разделенные шумовыми интервалами.

## 5. Что в итоге получается

`dataset\_simul`: набор из 100 коротких ERP-подобных сегментов `(пример, канал, время) = (100, 8, 400)` с P300 и N100, + шум.

`targets`: метки вида `[буква, номер\_прогона]` для каждого примера.

`new\_dataset\_simul`: один длинный многоканальный сигнал `(8, ~60k)`, собранный из 30 «задач» (каждая — 3–6 эпизодов одной буквы), эпизоды разделены случайными шумовыми вставками.