```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": [],
      "toc_visible": true
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "# Basic Python"
      ],
      "metadata": {
        "id": "McSxJAwcOdZ1"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 1. Split this string"
      ],
      "metadata": {
        "id": "CU48hgo4Owz5"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "s = \"Hi there Sam!\""
      ],
      "metadata": {
        "id": "s07c7JK7Oqt-"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "s=s.split()"
      ],
      "metadata": {
        "id": "6mGVa3SQYLkb"
      },
      "execution_count": null,
      "outputs": []
    },
```

```
    {
      "cell_type": "code",
      "source": [
        "print(s)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "rAHV0loaTBiU",
        "outputId": "aebf389a-5477-4dad-a7a5-5bf0d225e0e0"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "['Hi', 'there', 'Sam!']\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 2. Use .format() to print the following string. \n",
        "\n",
        "### Output should be: The diameter of Earth is 12742
kilometers."
      ],
      "metadata": {
        "id": "GH1QBn8HP375"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "planet = \"Earth\"\n",
        "diameter = 12742"
      ],
      "metadata": {
        "id": "_ZHoml3kPqic"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "planet = 'The diameter of the {} is {}
kilometers'.format(planet,diameter)"
      ],
      "metadata": {
        "id": "HyRyJv6CYPb4"
      },
      "execution_count": null,
      "outputs": []
```

```
    },
    {
      "cell_type": "code",
      "source": [
        "print(planet)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "5g3H2iEYUtuP",
        "outputId": "dc0c08a8-76dc-4337-fc03-fdb585d02e35"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "The diameter of the Earth is 12742 kilometers\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 3. In this nest dictionary grab the word \"hello\""
      ],
      "metadata": {
        "id": "KE74ZEwkRExZ"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "d =
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]}]}"
      ],
      "metadata": {
        "id": "fcVwbCc1QrQI"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print(d['k1'][3]['tricky'][3]['target'][3])"
      ],
      "metadata": {
        "id": "MvbkMZpXYRaw",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "cb24900b-625b-4125-fffd-0ef320efd075"
      },
```

```
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "hello\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# Numpy"
  ],
  "metadata": {
    "id": "bw0vVp-9ddjv"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ],
  "metadata": {
    "id": "LLiE_TYrhA1O"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
  ],
  "metadata": {
    "id": "wOg8hinbgx30"
  }
},
{
  "cell_type": "code",
  "source": [
    "array=np.zeros(10)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(array)"
  ],
  "metadata": {
```

```
    "id": "e4005lsTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "8696538c-2282-4dcd-c1ca-ade49258936c"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "array=np.ones(10)*5"
  ],
  "metadata": {
    "id": "raSO99YWXoPD"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(array)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "bAiuKfucXxa1",
    "outputId": "9ea09337-7484-4a61-e688-793e89ffd7cf"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ],
  "metadata": {
    "id": "gZHHDUBvrMX4"
```

```
  }
},
{
  "cell_type": "code",
  "source": [
    "array=np.arange(20,35,2)"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(array)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "2LzJjOsbYkEd",
    "outputId": "aee513c6-6a70-4409-b415-ba4fc373f91a"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[20 22 24 26 28 30 32 34]\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
  ],
  "metadata": {
    "id": "NaOM308NsRpZ"
  }
},
{
  "cell_type": "code",
  "source": [
    "x =  np.arange(0,9).reshape(3,3)"
  ],
  "metadata": {
    "id": "tOlEVH7BYceE"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
```

```
    "source": [
      "print(x)"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "V0KjsBmHaM6_",
      "outputId": "507699d6-3496-437c-e9b6-8a39c3f62f22"
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "[[0 1 2]\n",
          " [3 4 5]\n",
          " [6 7 8]]\n"
        ]
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "## 7. Concatenate a and b \n",
      "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
    ],
    "metadata": {
      "id": "hQ0dnhAQuU_p"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "a=([1, 2,3])"
    ],
    "metadata": {
      "id": "rAPSw97aYfE0"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "b=([4,5,6])"
    ],
    "metadata": {
      "id": "qZDTBvD_bGqh"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
```

```
    "x=np.concatenate((a, b))"
   ],
   "metadata": {
    "id": "tULcZ4BMbQj4"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "print(x)"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "BzmHxdx2bUIr",
    "outputId": "9e4ddc56-8384-4b20-f917-974434911fcb"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[1 2 3 4 5 6]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# Pandas"
   ],
   "metadata": {
    "id": "dlPEY9DRwZga"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
   ],
   "metadata": {
    "id": "ijoYW51zwr87"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "import pandas as pd"
   ],
   "metadata": {
    "id": "T5OxJRZ8uvR7"
   },
   "execution_count": null,
```

```
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "A=[]"
      ],
      "metadata": {
        "id": "xNpI_XXoYhs0"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "f=pd.DataFrame(A, columns=['a', 'b'], index=['1', '2', '3'])"
      ],
      "metadata": {
        "id": "V06ifOgwdiZe"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print(f)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "zZxbJ0F3dl4f",
        "outputId": "183718b5-e7e7-45e2-a54d-bf6348d5a230"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "     a    b\n",
            "1  NaN  NaN\n",
            "2  NaN  NaN\n",
            "3  NaN  NaN\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 9. Generate the series of dates from 1st Jan, 2023 to 10th
Feb, 2023"
      ],
      "metadata": {
        "id": "UXSmdNclyJQD"
```

```
  }
},
{
  "cell_type": "code",
  "source": [
    "per1 = pd.date_range(start ='1-1-2023',end ='10-02-2023')"
  ],
  "metadata": {
    "id": "dgyC0JhVYl4F"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "for val in per1:\n",
    "    print(val)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "5PbXb5kuexMu",
    "outputId": "198e35c8-fa95-4ec0-dc26-59dda3aa20c9"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "2023-01-01 00:00:00\n",
        "2023-01-02 00:00:00\n",
        "2023-01-03 00:00:00\n",
        "2023-01-04 00:00:00\n",
        "2023-01-05 00:00:00\n",
        "2023-01-06 00:00:00\n",
        "2023-01-07 00:00:00\n",
        "2023-01-08 00:00:00\n",
        "2023-01-09 00:00:00\n",
        "2023-01-10 00:00:00\n",
        "2023-01-11 00:00:00\n",
        "2023-01-12 00:00:00\n",
        "2023-01-13 00:00:00\n",
        "2023-01-14 00:00:00\n",
        "2023-01-15 00:00:00\n",
        "2023-01-16 00:00:00\n",
        "2023-01-17 00:00:00\n",
        "2023-01-18 00:00:00\n",
        "2023-01-19 00:00:00\n",
        "2023-01-20 00:00:00\n",
        "2023-01-21 00:00:00\n",
        "2023-01-22 00:00:00\n",
        "2023-01-23 00:00:00\n",
        "2023-01-24 00:00:00\n",
        "2023-01-25 00:00:00\n",
        "2023-01-26 00:00:00\n",
```

"2023-01-27 00:00:00\n",
"2023-01-28 00:00:00\n",
"2023-01-29 00:00:00\n",
"2023-01-30 00:00:00\n",
"2023-01-31 00:00:00\n",
"2023-02-01 00:00:00\n",
"2023-02-02 00:00:00\n",
"2023-02-03 00:00:00\n",
"2023-02-04 00:00:00\n",
"2023-02-05 00:00:00\n",
"2023-02-06 00:00:00\n",
"2023-02-07 00:00:00\n",
"2023-02-08 00:00:00\n",
"2023-02-09 00:00:00\n",
"2023-02-10 00:00:00\n",
"2023-02-11 00:00:00\n",
"2023-02-12 00:00:00\n",
"2023-02-13 00:00:00\n",
"2023-02-14 00:00:00\n",
"2023-02-15 00:00:00\n",
"2023-02-16 00:00:00\n",
"2023-02-17 00:00:00\n",
"2023-02-18 00:00:00\n",
"2023-02-19 00:00:00\n",
"2023-02-20 00:00:00\n",
"2023-02-21 00:00:00\n",
"2023-02-22 00:00:00\n",
"2023-02-23 00:00:00\n",
"2023-02-24 00:00:00\n",
"2023-02-25 00:00:00\n",
"2023-02-26 00:00:00\n",
"2023-02-27 00:00:00\n",
"2023-02-28 00:00:00\n",
"2023-03-01 00:00:00\n",
"2023-03-02 00:00:00\n",
"2023-03-03 00:00:00\n",
"2023-03-04 00:00:00\n",
"2023-03-05 00:00:00\n",
"2023-03-06 00:00:00\n",
"2023-03-07 00:00:00\n",
"2023-03-08 00:00:00\n",
"2023-03-09 00:00:00\n",
"2023-03-10 00:00:00\n",
"2023-03-11 00:00:00\n",
"2023-03-12 00:00:00\n",
"2023-03-13 00:00:00\n",
"2023-03-14 00:00:00\n",
"2023-03-15 00:00:00\n",
"2023-03-16 00:00:00\n",
"2023-03-17 00:00:00\n",
"2023-03-18 00:00:00\n",
"2023-03-19 00:00:00\n",
"2023-03-20 00:00:00\n",
"2023-03-21 00:00:00\n",
"2023-03-22 00:00:00\n",
"2023-03-23 00:00:00\n",
"2023-03-24 00:00:00\n",
"2023-03-25 00:00:00\n",

```
"2023-03-26 00:00:00\n",
"2023-03-27 00:00:00\n",
"2023-03-28 00:00:00\n",
"2023-03-29 00:00:00\n",
"2023-03-30 00:00:00\n",
"2023-03-31 00:00:00\n",
"2023-04-01 00:00:00\n",
"2023-04-02 00:00:00\n",
"2023-04-03 00:00:00\n",
"2023-04-04 00:00:00\n",
"2023-04-05 00:00:00\n",
"2023-04-06 00:00:00\n",
"2023-04-07 00:00:00\n",
"2023-04-08 00:00:00\n",
"2023-04-09 00:00:00\n",
"2023-04-10 00:00:00\n",
"2023-04-11 00:00:00\n",
"2023-04-12 00:00:00\n",
"2023-04-13 00:00:00\n",
"2023-04-14 00:00:00\n",
"2023-04-15 00:00:00\n",
"2023-04-16 00:00:00\n",
"2023-04-17 00:00:00\n",
"2023-04-18 00:00:00\n",
"2023-04-19 00:00:00\n",
"2023-04-20 00:00:00\n",
"2023-04-21 00:00:00\n",
"2023-04-22 00:00:00\n",
"2023-04-23 00:00:00\n",
"2023-04-24 00:00:00\n",
"2023-04-25 00:00:00\n",
"2023-04-26 00:00:00\n",
"2023-04-27 00:00:00\n",
"2023-04-28 00:00:00\n",
"2023-04-29 00:00:00\n",
"2023-04-30 00:00:00\n",
"2023-05-01 00:00:00\n",
"2023-05-02 00:00:00\n",
"2023-05-03 00:00:00\n",
"2023-05-04 00:00:00\n",
"2023-05-05 00:00:00\n",
"2023-05-06 00:00:00\n",
"2023-05-07 00:00:00\n",
"2023-05-08 00:00:00\n",
"2023-05-09 00:00:00\n",
"2023-05-10 00:00:00\n",
"2023-05-11 00:00:00\n",
"2023-05-12 00:00:00\n",
"2023-05-13 00:00:00\n",
"2023-05-14 00:00:00\n",
"2023-05-15 00:00:00\n",
"2023-05-16 00:00:00\n",
"2023-05-17 00:00:00\n",
"2023-05-18 00:00:00\n",
"2023-05-19 00:00:00\n",
"2023-05-20 00:00:00\n",
"2023-05-21 00:00:00\n",
"2023-05-22 00:00:00\n",
```

```
"2023-05-23 00:00:00\n",
"2023-05-24 00:00:00\n",
"2023-05-25 00:00:00\n",
"2023-05-26 00:00:00\n",
"2023-05-27 00:00:00\n",
"2023-05-28 00:00:00\n",
"2023-05-29 00:00:00\n",
"2023-05-30 00:00:00\n",
"2023-05-31 00:00:00\n",
"2023-06-01 00:00:00\n",
"2023-06-02 00:00:00\n",
"2023-06-03 00:00:00\n",
"2023-06-04 00:00:00\n",
"2023-06-05 00:00:00\n",
"2023-06-06 00:00:00\n",
"2023-06-07 00:00:00\n",
"2023-06-08 00:00:00\n",
"2023-06-09 00:00:00\n",
"2023-06-10 00:00:00\n",
"2023-06-11 00:00:00\n",
"2023-06-12 00:00:00\n",
"2023-06-13 00:00:00\n",
"2023-06-14 00:00:00\n",
"2023-06-15 00:00:00\n",
"2023-06-16 00:00:00\n",
"2023-06-17 00:00:00\n",
"2023-06-18 00:00:00\n",
"2023-06-19 00:00:00\n",
"2023-06-20 00:00:00\n",
"2023-06-21 00:00:00\n",
"2023-06-22 00:00:00\n",
"2023-06-23 00:00:00\n",
"2023-06-24 00:00:00\n",
"2023-06-25 00:00:00\n",
"2023-06-26 00:00:00\n",
"2023-06-27 00:00:00\n",
"2023-06-28 00:00:00\n",
"2023-06-29 00:00:00\n",
"2023-06-30 00:00:00\n",
"2023-07-01 00:00:00\n",
"2023-07-02 00:00:00\n",
"2023-07-03 00:00:00\n",
"2023-07-04 00:00:00\n",
"2023-07-05 00:00:00\n",
"2023-07-06 00:00:00\n",
"2023-07-07 00:00:00\n",
"2023-07-08 00:00:00\n",
"2023-07-09 00:00:00\n",
"2023-07-10 00:00:00\n",
"2023-07-11 00:00:00\n",
"2023-07-12 00:00:00\n",
"2023-07-13 00:00:00\n",
"2023-07-14 00:00:00\n",
"2023-07-15 00:00:00\n",
"2023-07-16 00:00:00\n",
"2023-07-17 00:00:00\n",
"2023-07-18 00:00:00\n",
"2023-07-19 00:00:00\n",
```

"2023-07-20 00:00:00\n",
"2023-07-21 00:00:00\n",
"2023-07-22 00:00:00\n",
"2023-07-23 00:00:00\n",
"2023-07-24 00:00:00\n",
"2023-07-25 00:00:00\n",
"2023-07-26 00:00:00\n",
"2023-07-27 00:00:00\n",
"2023-07-28 00:00:00\n",
"2023-07-29 00:00:00\n",
"2023-07-30 00:00:00\n",
"2023-07-31 00:00:00\n",
"2023-08-01 00:00:00\n",
"2023-08-02 00:00:00\n",
"2023-08-03 00:00:00\n",
"2023-08-04 00:00:00\n",
"2023-08-05 00:00:00\n",
"2023-08-06 00:00:00\n",
"2023-08-07 00:00:00\n",
"2023-08-08 00:00:00\n",
"2023-08-09 00:00:00\n",
"2023-08-10 00:00:00\n",
"2023-08-11 00:00:00\n",
"2023-08-12 00:00:00\n",
"2023-08-13 00:00:00\n",
"2023-08-14 00:00:00\n",
"2023-08-15 00:00:00\n",
"2023-08-16 00:00:00\n",
"2023-08-17 00:00:00\n",
"2023-08-18 00:00:00\n",
"2023-08-19 00:00:00\n",
"2023-08-20 00:00:00\n",
"2023-08-21 00:00:00\n",
"2023-08-22 00:00:00\n",
"2023-08-23 00:00:00\n",
"2023-08-24 00:00:00\n",
"2023-08-25 00:00:00\n",
"2023-08-26 00:00:00\n",
"2023-08-27 00:00:00\n",
"2023-08-28 00:00:00\n",
"2023-08-29 00:00:00\n",
"2023-08-30 00:00:00\n",
"2023-08-31 00:00:00\n",
"2023-09-01 00:00:00\n",
"2023-09-02 00:00:00\n",
"2023-09-03 00:00:00\n",
"2023-09-04 00:00:00\n",
"2023-09-05 00:00:00\n",
"2023-09-06 00:00:00\n",
"2023-09-07 00:00:00\n",
"2023-09-08 00:00:00\n",
"2023-09-09 00:00:00\n",
"2023-09-10 00:00:00\n",
"2023-09-11 00:00:00\n",
"2023-09-12 00:00:00\n",
"2023-09-13 00:00:00\n",
"2023-09-14 00:00:00\n",
"2023-09-15 00:00:00\n",

```
        "2023-09-16 00:00:00\n",
        "2023-09-17 00:00:00\n",
        "2023-09-18 00:00:00\n",
        "2023-09-19 00:00:00\n",
        "2023-09-20 00:00:00\n",
        "2023-09-21 00:00:00\n",
        "2023-09-22 00:00:00\n",
        "2023-09-23 00:00:00\n",
        "2023-09-24 00:00:00\n",
        "2023-09-25 00:00:00\n",
        "2023-09-26 00:00:00\n",
        "2023-09-27 00:00:00\n",
        "2023-09-28 00:00:00\n",
        "2023-09-29 00:00:00\n",
        "2023-09-30 00:00:00\n",
        "2023-10-01 00:00:00\n",
        "2023-10-02 00:00:00\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    "         [2, 'bbb', 25],\n",
    "         [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "ZizSetD-y5az"
  }
},
{
  "cell_type": "code",
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "_XMC8aEt0llB"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "l= pd.DataFrame(lists)"
  ],
  "metadata": {
    "id": "knH76sDKYsVX"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
```

```
      "source": [
        "print(l)"
      ],
      "metadata": {
        "id": "uLhHvxsIf2Mn",
        "outputId": "c7b79d16-b458-4e41-b546-b43cfea7a9d4",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "   0    1   2\n",
            "0  1  aaa  22\n",
            "1  2  bbb  25\n",
            "2  3  ccc  24\n"
          ]
        }
      ]
    }
  ]
}
```