```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 12,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "kaixGjVsyjgq",
        "outputId": "aa0375cb-de30-4b02-dfaa-6769c786441c"
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[1 2 3 4 5]\n",
            "[[0 0 0]\n",
            " [0 0 0]\n",
            " [0 0 0]]\n",
            "[[1 1]\n",
            " [1 1]]\n",
            "[0 1 2 3 4 5 6 7 8 9]\n",
            "[[1]\n",
            " [2]\n",
            " [3]\n",
            " [4]\n",
            " [5]]\n",
            "[3 4]\n",
            "[1 2 3 4]\n",
            "[5 6 7 8]\n"
          ]
        }
      ],
      "source": [
        "import numpy as np\n",
        "arr=np.array([1,2,3,4,5])\n",
        "print(arr)\n",
        "zeros_arr=np.zeros((3,3),dtype=int)\n",
        "print(zeros_arr)\n",
        "ones_arr=np.ones((2,2),dtype=int)\n",
        "print(ones_arr)\n",
        "arange_arr=np.arange(10)\n",
```

```
    "print(arange_arr)\n",
    "//array manipulation\n",
    "reshaped_arr=arr.reshape(5,1)\n",
    "print(reshaped_arr)\n",
    "sliced_arr=arr[2:4]\n",
    "print(sliced_arr)\n",
    "\n",
    ""
  ]
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "a=np.array([[4,5,6,7],[8,6,9,10]])\n",
    "b=a.T\n",
    "print(b)"
  ],
  "metadata": {
    "id": "mnBuXLm0CnAj",
    "outputId": "14477647-a0f8-4180-af1d-8654b456c23e",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": 29,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[[ 4  8]\n",
        " [ 5  6]\n",
        " [ 6  9]\n",
        " [ 7 10]]\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "a=np.array([1,2,3,4])\n",
    "b=np.array([5,6,7,8])\n",
    "c=np.stack(a+b)\n",
    "print(c)\n",
    "d=np.split(a,2)\n",
    "print(d)\n"
  ],
  "metadata": {
    "id": "YWlVlLY_Co_z",
    "outputId": "6652df69-4ae9-487d-f868-60dd09cd3a1b",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": 32,
```

```
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[ 6  8 10 12]\n",
            "[array([1, 2]), array([3, 4])]\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "import numpy as np\n",
        "#linear algebra with numpy\n",
        "a=np.array([[1,2],[3,4]])\n",
        "b=np.array([[7,8],[9,6]])\n",
        "c=np.dot(a,b)\n",
        "print(c)\n",
        "d=np.linalg.eig(c)\n",
        "print(d)"
      ],
      "metadata": {
        "id": "BORa_JL7Ja5G",
        "outputId": "dca28278-73fe-46f5-8beb-5c8fecbd8e2b",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": 42,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[[25 20]\n",
            " [57 48]]\n",
            "(array([ 0.83138635, 72.16861365]), array([[-0.63753723, -
0.39036909],\n",
            "       [ 0.77041955, -0.92065844]]))\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "a1=np.array([1,2,3,4])\n",
        "//broadcasting\n",
        "a11=a1+3\n",
        "print(a11)\n",
        "print(a1)\n",
        "a2=np.array([5,6,7,8])\n",
        "print(a2)\n",
        "sum=a1+a2\n",
        "print(sum)"
      ],
```

```
      "metadata": {
        "id": "gKtrCt_TA9SS",
        "outputId": "d7d0c839-4028-4d38-af03-22af85d33309",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": 14,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[4 5 6 7]\n",
            "[1 2 3 4]\n",
            "[5 6 7 8]\n",
            "[ 6  8 10 12]\n"
          ]
        }
      ]
    }
  ]
}
```