|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | {    "nbformat": 4,    "nbformat\_minor": 0,    "metadata": {      "colab": {        "provenance": [],        "collapsed\_sections": [          "wOg8hinbgx30",          "gZHHDUBvrMX4",          "NaOM308NsRpZ",          "hQ0dnhAQuU\_p",          "UXSmdNclyJQD",          "ZizSetD-y5az"        ]      },      "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!\"\n",          "a=s.split(\" \")\n",          "a\n"        ],        "metadata": {          "id": "s07c7JK7Oqt-",          "outputId": "6347fd14-46a1-4840-abfd-c86192e4757b",          "colab": {            "base\_uri": "<https://localhost:8080/>"          }        },        "execution\_count": null,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "['Hi', 'there', 'Sam!']"              ]            },            "metadata": {},            "execution\_count": 3          }        ]      },      {        "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\n",          "\"The diameter of {} is {} kilometers\".format(planet,diameter)"        ],        "metadata": {          "id": "\_ZHoml3kPqic",          "colab": {            "base\_uri": "<https://localhost:8080/>",            "height": 36          },          "outputId": "a99d365a-bc9e-4639-ce00-da924260fc03"        },        "execution\_count": 2,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "'The diameter of Earth is 12742 kilometers'"              ],              "application/vnd.google.colaboratory.intrinsic+json": {                "type": "string"              }            },            "metadata": {},            "execution\_count": 2          }        ]      },      {        "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']}]}]}\n",          "print(d['k1'][3][\"tricky\"][3]['target'][3])"        ],        "metadata": {          "id": "fcVwbCc1QrQI",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "4dca2f68-8435-4e01-f4de-ed3cea19a614"        },        "execution\_count": 17,        "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": [          "import numpy as np\n",          "array=np.zeros(10)\n",          "array"        ],        "metadata": {          "id": "NHrirmgCYXvU",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "4101420f-eb30-4ec9-8bcc-fc931e7428c5"        },        "execution\_count": 15,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"              ]            },            "metadata": {},            "execution\_count": 15          }        ]      },      {        "cell\_type": "code",        "source": [          "import numpy as np\n",          "array=np.ones(10)\*5\n",          "array\n"        ],        "metadata": {          "id": "e4005lsTYXxx",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "c944fc1e-ec1f-4097-d008-8e5aa1bccded"        },        "execution\_count": 16,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"              ]            },            "metadata": {},            "execution\_count": 16          }        ]      },      {        "cell\_type": "markdown",        "source": [          "## 5. Create an array of all the even integers from 20 to 35"        ],        "metadata": {          "id": "gZHHDUBvrMX4"        }      },      {        "cell\_type": "code",        "source": [          "import numpy as np\n",          "array=np.arange(20,35,2)\n",          "print(array)"        ],        "metadata": {          "id": "oAI2tbU2Yag-",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "cbb736c7-b1b9-4398-a73f-f2b3649bc1e2"        },        "execution\_count": 14,        "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": [          "import numpy as np\n",          "a=np.arange(0,9).reshape(3,3)\n",          "a"        ],        "metadata": {          "id": "tOlEVH7BYceE",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "d2ab2593-0f77-4cea-f9ff-5f14aa8e5dc1"        },        "execution\_count": 13,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "array([[0, 1, 2],\n",                "       [3, 4, 5],\n",                "       [6, 7, 8]])"              ]            },            "metadata": {},            "execution\_count": 13          }        ]      },      {        "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": [          "import numpy as np\n",          "a = np.array([1, 2, 3])\n",          "b = np.array([4, 5, 6])\n",          "c=np.concatenate((a,b),axis=None)\n",          "c"        ],        "metadata": {          "id": "rAPSw97aYfE0",          "colab": {            "base\_uri": "<https://localhost:8080/>"          },          "outputId": "9f21afc7-2545-43d4-ce68-c186ab920688"        },        "execution\_count": 12,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "array([1, 2, 3, 4, 5, 6])"              ]            },            "metadata": {},            "execution\_count": 12          }        ]      },      {        "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\n",          "a={\"names\":[\"rahul\",\"shijo\",\"berin\"],\"ages\":[21,21,22]}\n",          "b=pd.DataFrame(a,index=[1,2,3])\n",          "b\n"        ],        "metadata": {          "id": "T5OxJRZ8uvR7",          "colab": {            "base\_uri": "<https://localhost:8080/>",            "height": 143          },          "outputId": "beae55aa-365a-4153-e6a6-ec7185826220"        },        "execution\_count": 10,        "outputs": [          {            "output\_type": "execute\_result",            "data": {              "text/plain": [                "   names  ages\n",                "1  rahul    21\n",                "2  shijo    21\n",                "3  berin    22"              ],              "text/html": [                "\n",                "  <div id=\"df-25da7c7f-d980-43d7-a244-780427e769d4\">\n",                "    <div class=\"colab-df-container\">\n",                "      <div>\n",                "<style scoped>\n",                "    .dataframe tbody tr th:only-of-type {\n",                "        vertical-align: middle;\n",                "    }\n",                "\n",                "    .dataframe tbody tr th {\n",                "        vertical-align: top;\n",                "    }\n",                "\n",                "    .dataframe thead th {\n",                "        text-align: right;\n",                "    }\n",                "</style>\n",                "<table border=\"1\" class=\"dataframe\">\n",                "  <thead>\n",                "    <tr style=\"text-align: right;\">\n",                "      <th></th>\n",                "      <th>names</th>\n",                "      <th>ages</th>\n",                "    </tr>\n",                "  </thead>\n",                "  <tbody>\n",                "    <tr>\n",                "      <th>1</th>\n",                "      <td>rahul</td>\n",                "      <td>21</td>\n",                "    </tr>\n",                "    <tr>\n",                "      <th>2</th>\n",                "      <td>shijo</td>\n",                "      <td>21</td>\n",                "    </tr>\n",                "    <tr>\n",                "      <th>3</th>\n",                "      <td>berin</td>\n",                "      <td>22</td>\n",                "    </tr>\n",                "  </tbody>\n",                "</table>\n",                "</div>\n",                "      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-25da7c7f-d980-43d7-a244-780427e769d4')\"\n",                "              title=\"Convert this dataframe to an interactive table.\"\n",                "              style=\"display:none;\">\n",                "        \n",                "  <svg xmlns=\"[http://www.w3.org/2000/svg\](http://www.w3.org/2000/svg%5C)" height=\"24px\"viewBox=\"0 0 24 24\"\n",                "       width=\"24px\">\n",                "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",                "    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4   |  |  | | --- | --- | | https://ssl.gstatic.com/ui/v1/icons/mail/no_photo.png |  | |  |