

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "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": 1,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "x = s.split()\n",
        "\n",
        "print(x)"
      ],
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      }
    }
  ]
}

```



```

    },
    "outputId": "d6a038ab-6b07-4fce-dd64-c332d3fb3b5d"
  },
  "execution_count": 2,
  "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": [
    "print(\"The          diameter          of          {planet}          is          {diameter} kilometers.\").format(planet=\"Earth\",diameter=12742))"
  ],
  "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "f712157f-fe9b-46d8-94b5-1be0d18b00ba"
},
"execution_count": 6,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "The diameter of 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']}]}]}\n",
    "d['k1'][3]['tricky'][3]['target'][3]"
  ],
  "metadata": {
    "id": "fcVwbCc1QrQI",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 35
    },
    "outputId": "2f177b00-46c4-4348-891a-c61e603bfec4"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 8
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# Numpy"
  ],
  "metadata": {
    "id": "bw0vVp-9ddjv"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ]
}

```



```

],
"metadata": {
  "id": "LLiE_TYrhA10"
},
"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",
    "np.zeros(10)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "45b49383-d7c4-4ca5-eb21-50a2c553e64c"
  },
  "execution_count": 10,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
        ]
      },
      "metadata": {},
      "execution_count": 10
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "np.ones(10) * 5"
  ],
  "metadata": {
    "id": "e4005lsTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "a4d484af-11f7-453a-a07d-0632eb4f8088"
  },

```



```

"execution_count": 11,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
      ]
    },
    "metadata": {},
    "execution_count": 11
  }
],
{
  "cell_type": "markdown",
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ],
  "metadata": {
    "id": "gZHDUBvrMX4"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "print(np.arange(20,35,2))"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "9cad3feb-1012-4568-cbaf-2b7bfe5f4f67"
  },
  "execution_count": 13,
  "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",
    "np.arange(0,9).reshape((3,3))"
],
"metadata": {
    "id": "tOIEVH7BYceE",
    "colab": {
        "base_uri": "https://localhost:8080/"
    },
    "outputId": "a4a8dca9-6925-46c4-9411-7c0fed043e8e"
},
"execution_count": 14,
"outputs": [
    {
        "output_type": "execute_result",
        "data": {
            "text/plain": [
                "array([[0, 1, 2],\n",
                "       [3, 4, 5],\n",
                "       [6, 7, 8]])"
            ]
        },
        "metadata": {},
        "execution_count": 14
    }
],
},
{
    "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",
        "np.concatenate((a, b), axis=None)"
    ],
    "metadata": {
        "id": "rAPSw97aYfE0",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "ff62527b-ae69-49b6-fd30-b9b7c8e35403"
    },
    "execution_count": 16,
    "outputs": [
        {
            "output_type": "execute_result",

```



```

        "data": {
          "text/plain": [
            "array([1, 2, 3, 4, 5, 6])"
          ]
        },
        "metadata": {},
        "execution_count": 16
      }
    ],
  },
  {
    "cell_type": "markdown",
    "source": [
      "# Pandas"
    ],
    "metadata": {
      "id": "dIPEY9DRwZga"
    }
  },
  {
    "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"
    ],
    "metadata": {
      "id": "T5OxJRZ8uvR7"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "import pandas as pd\n",
      "pd.DataFrame(index=np.arange(3), columns=np.arange(2))\n",
      "\n"
    ],
    "metadata": {
      "id": "xNpI_XXoYhs0",
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 143
      },
      "outputId": "b35ad381-1b37-4d7c-a9b4-149a226c446c"
    },
    "execution_count": 17,
    "outputs": [
      {

```



```

"output_type": "execute_result",
"data": {
  "text/plain": [
    "      0      1\n",
    "0  NaN  NaN\n",
    "1  NaN  NaN\n",
    "2  NaN  NaN"
  ],
  "text/html": [
    "\n",
    "  <div id=\"df-ee70e431-d84b-4fef-837b-1efd170613de\">\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>0</th>\n",
    "              <th>1</th>\n",
    "            </tr>\n",
    "          </thead>\n",
    "          <tbody>\n",
    "            <tr>\n",
    "              <th>0</th>\n",
    "              <td>NaN</td>\n",
    "              <td>NaN</td>\n",
    "            </tr>\n",
    "            <tr>\n",
    "              <th>1</th>\n",
    "              <td>NaN</td>\n",
    "              <td>NaN</td>\n",
    "            </tr>\n",
    "            <tr>\n",
    "              <th>2</th>\n",
    "              <td>NaN</td>\n",
    "              <td>NaN</td>\n",
    "            </tr>\n",
    "          </tbody>\n",
    "        </table>\n",
    "      </div>\n",
    "      <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ee70e431-d84b-4fef-837b-1efd170613de')\" title=\"Convert this dataframe to an interactive table.\" style=\"display: none;\">\n",

```




```

"      \n",
"      <svg xmlns=\"http://www.w3.org/2000/svg\" 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 2.06.94-2.06.94-2.06.94-2.06.94
2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06.94-2.06.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94
2.06.94-2.06.94-2.06.94-2.06.94 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 2.05 0
2.83L4 21.41c.39.39.95.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-
2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"      </svg>\n",
"      </button>\n",
"      \n",
"      <style>\n",
"      .colab-df-container {\n",
"      display: flex;\n",
"      flex-wrap: wrap;\n",
"      gap: 12px;\n",
"      }\n",
"      \n",
"      .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"      }\n",
"      \n",
"      .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"      }\n",
"      \n",
"      [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"      }\n",
"      \n",
"      [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"      }\n",
"      </style>\n",
"      \n",
"      <script>\n",
"      const buttonEl =\n",
"      document.querySelector('#df-ee70e431-d84b-4fef-837b-
1efd170613de button.colab-df-convert');\n",

```



```

        buttonEl.style.display = "\n",
        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
    "\n",
    "        async function convertToInteractive(key) {\n",
    "            const element = document.querySelector('#df-ee70e431-d84b-4fef-837b-1efd170613de');\n",
    "            const dataTable = \n",
    "
    google.colab.kernel.invokeFunction('convertToInteractive',\n",
    "
    [key], {});\n",
    "            if (!dataTable) return;\n",
    "
    "\n",
    "            const docLinkHtml = 'Like what you see? Visit the ' + \n",
    "
    ' <a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>\n",
    "
    + ' to learn more about interactive tables.';\n",
    "            element.innerHTML = \";\n",
    "            dataTable['output_type'] = 'display_data';\n",
    "            await google.colab.output.renderOutput(dataTable, element);\n",
    "            const docLink = document.createElement('div');\n",
    "            docLink.innerHTML = docLinkHtml;\n",
    "            element.appendChild(docLink);\n",
    "        }\n",
    "    </script>\n",
    "    </div>\n",
    "    </div>\n",
    "    "
    ]
    },
    "metadata": {},
    "execution_count": 17
  }
]
},
{
  "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": [
    "from datetime import date, timedelta\n",
    "\n",
    "sdate = date(2023,1,1) \n",
    "edate = date(2023,2,10) \n",
    "\n",
    "def dates_bwn_twodates(start_date, end_date):\n",
    "    for n in range(int ((end_date - start_date).days)):\n",
    "        yield start_date + timedelta(n)\n",
    "    print(dates_bwn_twodates(sdate,edate))"
  ],

```



```

"metadata": {
  "id": "dgyC0JhVYI4F",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "772903bd-cd5d-4080-a084-151bdc9bfafb"
},
"execution_count": 19,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "<generator object dates_bwn_twodates at 0x7f730f9fedd0>\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": "_XMC8aEt0IIB"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "\n",
    "import pandas as pd\n",
    "\n",
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
    "\n",
    "df = pd.DataFrame(lists, columns=['sno','name','number'])\n",
    "print(df)"
  ],
  "metadata": {
    "id": "knH76sDKYsVX",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  }
}

```



```

    },
    "outputId": "dd0dc288-79da-4f73-de4a-1056dda7c489"
  },
  "execution_count": 21,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "      sno name  number\n",
        "0      1  aaa    22\n",
        "1      2  bbb    25\n",
        "2      3  ccc    24\n"
      ]
    }
  ]
}

```

