

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "McSxJAwcOdZ1"
      },
      "source": [
        "# Basic Python"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "CU48hgo4Owz5"
      },
      "source": [
        "## 1. Split this string"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "s07c7JK7Oqt-",
        "outputId": "eec0d977-fe89-4a29-db0a-6d3c3b7d064b"
      },
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
```

```

    "text": [
        "['Hi', 'there', 'Sam!']\n"
    ]
}
],
"source": [
    "s = \"Hi there Sam!\"\n",
    "print(s.split())"
]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "YKRLoY86RhLh"
    },
    "source": []
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "GH1QBn8HP375"
    },
    "source": [
        "## 2. Use .format() to print the following string. \n",
        "\n",
        "### Output should be: The diameter of Earth is 12742 kilometers."
    ]
},
{
    "cell_type": "code",
    "execution_count": null,

```

```

"metadata": {
  "id": "_ZHoml3kPqic",
  "outputId": "ccf378ae-5e7e-4aa1-dd97-8f12df4dad60"
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "The diameter of Earth is 12742\n"
    ]
  }
],
"source": [
  "# planet = \"Earth\\n\",
  "# diameter = 12742\\n\",
  "\\n\",
  "res = \"The diameter of {planet} is {diameter}\\n\",
  "print(res.format(planet=\"Earth\\n\",diameter=12742))"
],
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "KE74ZEwkRExZ"
  },
  "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
  ]
},
{

```

```
"cell_type": "code",
"execution_count": null,
"metadata": {
  "id": "fcVwbCc1QrQI",
  "outputId": "d3320004-f5e4-460e-b9db-43ff2369a31b"
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "hello\n"
    ]
  }
],
"source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]}]\n",
  "print(d['k1'][3]['tricky'][3]['target'][3])"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "bw0vVp-9ddjv"
  },
  "source": [
    "# Numpy"
  ]
},
{
  "cell_type": "code",
```

```
"execution_count": null,
"metadata": {
  "id": "LLiE_TYrhA1O"
},
"outputs": [],
"source": [
  "import numpy as np"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "wOg8hinbgx30"
  },
  "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
  ]
},
{
  "cell_type": "code",
  "execution_count": 2,
  "metadata": {
    "id": "NHirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "098f1d85-d7dd-40f0-c776-0eb37fd6523a"
},
"outputs": [
  {
```

```
"output_type": "stream",
"name": "stdout",
"text": [
  "[0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
]
},
],
"source": [
  "import numpy as np\n",
  "\n",
  "arr = np.zeros(10)\n",
  "\n",
  "print(arr)"
]
},
{
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {
    "id": "e4005lsTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "51c79bed-900c-403e-a3b2-2b0c15198a81"
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
      ]
    }
  ]
}
```

```
]
}
],
"source": [
  "import numpy as np\n",
  "\n",
  "arr = np.ones(10)*5\n",
  "\n",
  "print(arr)"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "gZHHDUBvrMX4"
  },
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "metadata": {
    "id": "oAl2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "451e5b5c-5904-4002-ac95-e01de4f37e14"
},
"outputs": [
```

```

{
  "output_type": "stream",
  "name": "stdout",
  "text": [
    "[ True False  True False  True False  True False  True False  True False\n",
    " True False  True False]\n",
    "[20 22 24 26 28 30 32 34]\n"
  ]
}
],
"source": [
  "import numpy as np\n",
  "x=np.arange(20,36)\n",
  "y=(x%2==0)\n",
  "print(y)\n",
  "z=x[y]\n",
  "print(z)"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "NaOM308NsRpZ"
  },
  "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
  ]
},
{
  "cell_type": "code",
  "execution_count": 17,

```



```
"metadata": {
  "id": "tOIEVH7BYceE",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "8bf4a793-7da9-465c-bb49-5b09277c2ddd"
},
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "[[0 1 2]\n",
      " [3 4 5]\n",
      " [6 7 8]]\n"
    ]
  }
],
"source": [
  "import numpy as np\n",
  "arr = np.arange(0,9).reshape((3,3))\n",
  "print(arr)"
],
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "hQ0dnhAQuU_p"
  },
  "source": [
    "## 7. Concatenate a and b \n",
```

```
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ]
},
{
  "cell_type": "code",
  "source": [],
  "metadata": {
    "id": "8uqz0UOUT_Uy"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "execution_count": 19,
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "d241ebd0-9ecf-41ca-ce51-9708ea7ab21f"
},
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "[1 2 3 4 5 6]\n"
    ]
  }
],
```

```
"source": [  
    "import numpy as np\n",  
    "\n",  
    "a = np.array([1,2,3])\n",  
    "b = np.array([4,5,6])\n",  
    "print(np.concatenate((a,b)))"  
]  
,  
{  
    "cell_type": "markdown",  
    "metadata": {  
        "id": "dIPEY9DRwZga"  
    },  
    "source": [  
        "# Pandas"  
    ]  
},  
{  
    "cell_type": "markdown",  
    "metadata": {  
        "id": "ijoYW51zwr87"  
    },  
    "source": [  
        "## 8. Create a dataframe with 3 rows and 2 columns"  
    ]  
},  
{  
    "cell_type": "code",  
    "execution_count": 20,  
    "metadata": {  
        "id": "T5OxJRZ8uvR7"
```

```

},
"outputs": [],
"source": [
    "import pandas as pd\n"
]
},
{
    "cell_type": "code",
    "execution_count": 27,
    "metadata": {
        "id": "xNpl_XXoYhs0",
        "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 143
        },
        "outputId": "6074f013-c300-4e4d-a216-d53fe6cebb57"
    },
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "  Alphabets ASCII\n",
                    "0      A   65\n",
                    "1      B   66\n",
                    "2      C   67"
                ]
            },
            "text/html": [
                "\n",
                " <div id=\"df-caa19c40-4b26-4f36-b767-32768b6ae31d\">\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>Alphabets</th>\n",
"            <th>ASCII</th>\n",
"        </tr>\n",
"    </thead>\n",
"    <tbody>\n",
"        <tr>\n",
"            <th>0</th>\n",
"            <td>A</td>\n",
"            <td>65</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>1</th>\n",
"            <td>B</td>
```

[illegible]

```

"\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-caa19c40-4b26-4f36-b767-32768b6ae31d 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-caa19c40-4b26-4f36-b767-
32768b6ae31d');\n",
"       const dataTable =\n",
"         await 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": 27
}
],
"source": [
  "data = [['A', 65], ['B', 66], ['C', 67]]\n",
  "\n",
  "df = pd.DataFrame(data, columns=['Alphabets', 'ASCII'])\n",
  "\n",
  "df"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "UXSmdNclyJQD"
  },
  "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
  ]
},
{
  "cell_type": "code",
  "execution_count": 33,
  "metadata": {
    "id": "dgyCOJhVYI4F",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },

```

```

"outputId": "fed1fe77-26e3-4de0-c5c6-929b32a13401"
},
"outputs": [
{
  "output_type": "execute_result",
  "data": {
    "text/plain": [
      "DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',\n",
      "                '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',\n",
      "                '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',\n",
      "                '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',\n",
      "                '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',\n",
      "                '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',\n",
      "                '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',\n",
      "                '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',\n",
      "                '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',\n",
      "                '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09'],\n",
      "                dtype='datetime64[ns]', freq='D')"
    ]
  },
  "metadata": {},
  "execution_count": 33
}
],
"source": [
  "from pandas.core.window.rolling import timedelta as td\n",
  "from pandas.io.sql import date as d\n",
  "import pandas as pd\n",
  "startDate = d(2023, 1, 1)\n",
  "endDate = d(2023, 2, 10)\n",
  "pd.date_range(startDate,endDate-td(days=1),freq='d')

```

```

]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "ZizSetD-y5az"
  },
  "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    "          [2, 'bbb', 25],\n",
    "          [3, 'ccc', 24]]"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "_XMC8aEt0IIB"
  },
  "outputs": [],
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ]
},
{
  "cell_type": "code",
  "execution_count": 37,
  "metadata": {
    "id": "knH76sDKYsVX",

```

```
"colab": {
  "base_uri": "https://localhost:8080/",
  "height": 143
},
"outputId": "458c7e90-5aae-432b-b111-edec7bd7a5d6"
},
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        " Number1 Letters Number2\n",
        "0    1   aaa    22\n",
        "1    2   bbb    25\n",
        "2    3   ccc    24"
      ],
      "text/html": [
        "\n",
        " <div id=\"df-3b0992a7-bc6f-4fb7-b0ee-3964bf184fec\">\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>Number1</th>\n",
"      <th>Letters</th>\n",
"      <th>Number2</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>aaa</td>\n",
"      <td>22</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>2</td>\n",
"      <td>bbb</td>\n",
"      <td>25</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3</td>\n",
"      <td>ccc</td>\n",
"      <td>24</td>\n",
"    </tr>\n",
```

[illegible]

```

" 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-3b0992a7-bc6f-4fb7-b0ee-3964bf184fec 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-3b0992a7-bc6f-4fb7-b0ee-3964bf184fec');\n",
  "        const dataTable =\n",
  "            await 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": 37
}
],
"source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",

```



```

    "df = pd.DataFrame(lists, columns=['Number1','Letters','Number2'])\n",
    "df"
]
}
],
"metadata": {
  "colab": {
    "collapsed_sections": [],
    "provenance": []
  },
  "kernel_spec": {
    "display_name": "Python 3.9.13 64-bit (microsoft store)",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.9.13"
  },
  "vscode": {
    "interpreter": {
      "hash": "33c99ce92ae57a37c90157b7c35934b668ba43f0729fdf7df53b30c6721ba8f2"
    }
  }
}

```

```
}  
},  
"nbformat": 4,  
"nbformat_minor": 0  
}
```