

```

{
  "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": 1,
      "metadata": {
        "id": "s07c7JK7Oqt-"
      },
      "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": "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": 3,
      "metadata": {
        "id": "_ZHoml3kPqic"
      },
      "outputs": [

```

```

{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "The diameter of Earth is 12742 kilometers\n"
  ]
},
{
  "source": [
    "msg = \"The diameter of {planet} is {diameter} kilometers\\n\\n\",
    "print(msg.format(planet=\"Earth\\", diameter=str(12742)))"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "KE74ZEwkRExZ"
  },
  "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
  ]
},
{
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    },
    "id": "MvbkMZpXYRaw",
    "outputId": "1b23bb20-476e-446f-bbe2-78b54bfd6954"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "'hello'"
        ]
      },
      "execution_count": 4,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "d =
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]]}]\n",
    "d['k1'][3]['tricky'][3]['target'][3]"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "bw0vVp-9ddjv"
  },
  "source": [

```

```

    "# Numpy"
  ]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "metadata": {
    "id": "LLiE_TYrhA10"
  },
  "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": 7,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "NHrirmgCYXvU",
    "outputId": "9d3d98dc-9a7d-48e5-f1f4-88522ea46050"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
        ]
      },
      "execution_count": 7,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "import numpy as np\n",
    "array=np.zeros(10)\n",
    "array"
  ]
},
{
  "cell_type": "code",
  "execution_count": 17,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  }
}

```

```

    },
    "id": "e40051sTYXxx",
    "outputId": "c5120875-1cd9-42fa-e5cf-9ff547ff08ca"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
        ]
      },
      "execution_count": 17,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "import numpy as np\n",
    "array=np.ones(10)*5\n",
    "array"
  ]
},
{
  "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": 8,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "oAI2tbU2Yag-",
    "outputId": "c13dba35-f3ff-4403-c767-4596779b11bd"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[20, 22, 24, 26, 28, 30, 32, 34]"
        ]
      },
      "execution_count": 8,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "arr=list(range(20,35,2))\n",
    "arr"
  ]
},

```

```

{
  "cell_type": "markdown",
  "metadata": {
    "id": "NaOM308NsRpZ"
  },
  "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
  ]
},
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "tO1EVH7BYceE",
    "outputId": "394f4a90-785c-4b22-9dbb-5813d01cb67f"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([[0, 1, 2],\n        [3, 4, 5],\n        [6, 7, 8]])"
        ]
      },
      "execution_count": 9,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "import numpy as np\n",
    "x = np.arange(0,9).reshape(3,3)\n",
    "x"
  ]
},
{
  "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",
  "execution_count": 10,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "rAPSw97aYfE0",
    "outputId": "c838987f-8690-40b4-c826-0886aa6d2c76"
  }
}

```

```

},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "[1 2 3 4 5 6]\n"
    ]
  }
],
"source": [
  "a = np.array([1, 2, 3])\n",
  "b = np.array([4, 5, 6])\n",
  "print(np.concatenate((a, b), axis=0))"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "dlPEY9DRwZga"
  },
  "source": [
    "# Pandas"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "ijoYW51zwr87"
  },
  "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 143
    },
    "id": "T5OxJRZ8uvR7",
    "outputId": "ecd09ec0-bafd-4e3e-c226-5f3deb2ea752"
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<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>Marks</th>\n",
"        </tr>\n",
"    </thead>\n",
"    <tbody>\n",
"        <tr>\n",
"            <th>0</th>\n",
"            <td>10</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>1</th>\n",
"            <td>20</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>2</th>\n",
"            <td>30</td>\n",
"        </tr>\n",
"    </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"    Marks\n",
"0      10\n",
"1      20\n",
"2      30"
]
},
"execution_count": 11,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"import pandas as pd\n",
"data = [10,20,30]\n",
"df = pd.DataFrame(data, columns=['Marks'])\n",
"df\n"
]
},
{
"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": 12,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "dgyC0JhVYl4F",
    "outputId": "cdfec08b-93ac-4e40-e53e-ba5a12144f52"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "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"
      ]
    }
  ]
}
```



```

],
"source": [
  "import pandas as pd\n",
  "perl = pd.date_range(start = '01-01-2023', \n",
  "    end = '02-10-2023', freq = 'D')\n",
  "    \n",
  "for val in perl:\n",
  "    print(val)"
]
},
{
  "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": 13,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 143
    },
    "id": "_XMC8aEt01lB",
    "outputId": "0cc3b986-e283-4dfe-fc6d-f5b0cace5299"
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "    .dataframe tbody tr th:only-of-type {\n",
          "        vertical-align: middle;\n",
          "    }\n",
          ".dataframe tbody tr th {\n",
          "    vertical-align: top;\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>Sno</th>\n",
          "      <th>Name</th>\n",

```

```

"      <th>Number</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",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"  Sno Name  Number\n",
"0     1   aaa     22\n",
"1     2   bbb     25\n",
"2     3   ccc     24"
]
},
"execution_count": 13,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
"df = pd.DataFrame(lists, columns =['Sno', 'Name', 'Number'])\n",
"df"
]
}
],
"metadata": {
"colab": {
"collapsed_sections": [],
"provenance": []
},
"kernel_spec": {
"display_name": "Python 3 (ipykernel)",
"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": "3.9.7"
},
"vscode": {
  "interpreter": {
    "hash":
"21e739e15d7fd7e70d380feb11aee9bcb95049acac1bc8b44c69aee560de935"
  }
}
},
"nbformat": 4,
"nbformat_minor": 1
}
```